

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Экономика программной инженерии  
Лабораторная работа № 1

Выполнили студенты:

Ефименко Кирилл

Ильин Артём

Группа № Р34202

Преподаватель: Машина Екатерина Алексеевна

г. Санкт-Петербург  
2024

<b>Задание.....</b>	<b>3</b>
<b>Вариант.....</b>	<b>3</b>
<b>Выполнение работы.....</b>	<b>3</b>
1. Набор функциональных требований.....	3
Регистрация и авторизация.....	3
Управление аккаунтом.....	4
Торговля.....	4
Спотовая торговля.....	4
Фьючерсная торговля.....	4
P2P-торговля.....	4
Пополнение и вывод средств.....	4
Инвестиции.....	5
Аналитика и отчеты.....	5
Образование и поддержка.....	5
Программы лояльности.....	5
Безопасность.....	6
Интеграция с мобильным приложением.....	6
2. Оценка трудоемкости разработки проекта наивным методом.....	7
3. Оценка трудоемкость разработки проекта.....	11
3.1 Метод PERT.....	11
3.2 Сетевая диаграмма взаимосвязи работ.....	14
3.3 Метод критического пути.....	14
Распределение времени по ролям:.....	15
Рабочий процесс:.....	15
4. Оценка размера проекта методом функциональных точек и расчет трудоёмкости методом COCOMO II.....	16
4.1 Метод функциональных точек.....	16
Подсчет функциональных точек, связанных с данными:.....	16
Расчет суммарного эффекта:.....	22
Расчет фактора выравнивания (VAF):.....	22
4.2 Метод COCOMO II.....	23
Оценка размера продукта в KSLOC.....	23
Оценка многокомпонентного продукта.....	25
5. Оценка размера методом Use Case Points.....	27
6. Сравнение результатов и выводы.....	32

## Задание

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 COSOMO II (Обновленная таблица количества строк на точку для разных языков программирования)
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.

## Вариант

Бинанс (<https://www.binance.com/en>)

## Выполнение работы

### 1. Набор функциональных требований

Хедер веб-проекта должен быть одинаковым на всех страницах.

Хедер должен содержать логотип креативного кластера, который при нажатии ведет на главную страницу. Также здесь должно быть меню со ссылками на все страницы проекта.

### Регистрация и авторизация

- Возможность регистрации с использованием электронной почты или номера телефона.
- Подтверждение аккаунта через двухфакторную аутентификацию (2FA) или SMS.
- Поддержка различных методов аутентификации (Google Authenticator, Biometric Login).

- Восстановление доступа к аккаунту через подтвержденную почту или номер телефона.

## **Управление аккаунтом**

- Просмотр и обновление персональных данных пользователя.
- Настройка уровня верификации (KYC) для увеличения лимитов операций.
- Управление настройками безопасности (включение/выключение 2FA, управление устройствами и активными сессиями).
- Поддержка разных уровней аккаунта (персональный, корпоративный).

## **Торговля**

### **Спотовая торговля**

- Покупка и продажа криптовалют на спотовом рынке.
- Просмотр и установка лимитных, рыночных и стоп-ордеров.
- Интерактивный график для анализа цен с использованием технических индикаторов.
- Подробный список доступных пар для торговли.

### **Фьючерсная торговля**

- Поддержка маржинальной торговли с использованием плеча.
- Создание рыночных и лимитных ордеров.
- Отслеживание уровня ликвидации.
- Гибкие настройки риск-менеджмента.

### **P2P-торговля**

- Покупка и продажа криптовалют через P2P-объявления.
- Фильтры для выбора продавца/покупателя по параметрам (валюта, страна, способ оплаты).
- Защита сделок с использованием эскроу-системы.

## **Пополнение и вывод средств**

- Поддержка множества криптовалютных кошельков для ввода и вывода.
- Автоматическое отображение текущих лимитов ввода/вывода.
- Поддержка банковских переводов и других платежных систем для работы с фиатом.

- История всех транзакций (депозиты, выводы, комиссии).

## **Инвестиции**

- Стейкинг криптовалют для получения пассивного дохода.
- Возможность участия в "Launchpad" для покупки токенов на стадии предварительного размещения.
- Сохранение средств на сберегательных продуктах (Flexible или Fixed Savings).
- Автоматическая настройка криптовалютных накоплений через программы "Auto-Invest"

## **Аналитика и отчеты**

- Доступ к рыночным данным в реальном времени: объем торгов, цены, глубина рынка.
- Просмотр детализированных отчетов по проведенным сделкам.
- Инструменты для построения графиков и анализа.
- Доступ к API для подключения сторонних аналитических систем.

## **Образование и поддержка**

- Интерактивные образовательные материалы о криптовалютах и торговле.
- Доступ к разделу помощи и FAQ.
- Круглосуточная служба поддержки через чат или электронную почту.
- Поддержка мультиязычности (включая русский и английский языки).

## **Программы лояльности**

- Реферальные программы с начислением бонусов за привлеченных пользователей.
- Интеграция системы скидок на торговые комиссии при использовании токена Binance Coin (BNB).
- Периодические промоакции и конкурсы для трейдеров.

## **Безопасность**

- Защита аккаунта с помощью биометрической аутентификации (на мобильных устройствах).
- Многоуровневая система защиты транзакций.
- Мониторинг подозрительной активности и временная блокировка аккаунтов.
- Использование SSL/TLS для передачи данных.

## **Интеграция с мобильным приложением**

- Синхронизация данных с веб-версией.
- Поддержка мобильного API для упрощения торговли и мониторинга.
- Уведомления о движении цен, ликвидации позиций и других важных событиях.

## 2. Оценка трудоемкости разработки проекта наивным методом

№	Задача	Краткое описание задачи	Оценка времени (часы)	Поправочный коэффициент	Трудоемкость с учетом риска (часы)
1	Проектирование решения		30	1.2	36
1.1	Сбор и анализ требований	Исследование бизнес-требований и функциональности платформы.	10	1.3	13
1.2	Проектирование архитектуры системы	Определение общей архитектуры, модулей и их взаимодействия.	10	1.2	12
1.3	Проектирование UI/UX	Создание макетов ключевых страниц и проработка пользовательских сценариев.	10	1.2	12
2	Разработка		260	1.3	338
2.1	Реализация модуля регистрации и авторизации	Вход, регистрация, восстановление доступа, настройка 2FA.	15	1.3	19.5
2.2	Создание базы данных	Разработка структуры таблиц и связей для хранения данных о пользователях, транзакциях и торговле.	12	1.2	14.4

2.3	Реализация спотовой торговли	Интерфейс для покупки/ продажи криптовалют, управление ордерами.	30	1.4	42
2.4	Реализация фьючерсной торговли	Поддержка маржинальной торговли, управление ордерами с использованием плеча.	30	1.4	42
2.5	P2P-платформа	Разработка функционала для торговли между пользователями с поддержкой эскроу.	20	1.4	28
2.6	Инвестиционные продукты	Функционал для стейкинга, Launchpad, сберегательных продуктов.	25	1.3	32.5
2.7	Модуль аналитики	Реализация графиков, технических индикаторов, таблиц цен.	15	1.3	19.5
2.8	Депозиты и вывод средств	Интеграция платежных шлюзов, поддержка криптовалютных кошельков.	15	1.4	21
2.9	API для интеграции	Создание публичного и приватного API для мобильных приложений и сторонних сервисов.	15	1.2	18



2.10	Мобильная адаптация интерфейса	Оптимизация интерфейсов под мобильные устройства.	20	1.3	26
2.11	Образовательные материалы	Создание раздела с интерактивными курсами, гайдами и видео.	10	1.2	12
2.12	Программы лояльности	Разработка реферальной системы и механики скидок на комиссии.	15	1.3	19.5
2.13	Функционал безопасности	Реализация мониторинга подозрительных действий, защиты аккаунта, 2FA.	20	1.4	28
<b>3</b>	<b>Тестирование</b>		50	1.2	60
3.1	Функциональное тестирование	Проверка корректности работы всех модулей.	20	1.2	24
3.2	Интеграционное тестирование	Проверка взаимодействия модулей и внешних сервисов.	15	1.3	19.5
3.3	Нагрузочное тестирование	Оценка производительности под высокой нагрузкой.	15	1.1	16.5
<b>4</b>	<b>Внедрение</b>		15	1.2	18
4.1	Развертывание на тестовом сервере	Установка и настройка проекта для внутреннего тестирования.	5	1.2	6

4.2	Финальное развертыван ие	Настройка на продакшн сервере, миграция данных.	10	1.2	12
<b>Итого</b>			<b>355</b>		<b>452.9</b>

### 3. Оценка трудоемкость разработки проекта

#### 3.1 Метод PERT

$$E_i = \frac{O + 4M + P}{6}$$

$$CKO_i = \frac{P - O}{6}$$

Где:

P - Pessimistic,

O - Optimistic,

M - Most likely (Optimal)

№	Задача	Краткое описание задачи	Optimistic (часы)	Pessimistic (часы)	Optimal (часы)	Ei	CKO i
1	Проектирование решения						
1.1	Сбор и анализ требований	Исследование бизнес-требований и функциональности платформы.	8	16	12	12	1.33
1.2	Проектирование архитектуры системы	Определение общей архитектуры, модулей и их взаимодействия.	10	30	20	20	3.33
1.3	Проектирование UI/UX	Создание макетов ключевых страниц и проработка пользовательских сценариев.	10	20	15	15	1.67
2	Разработка						

2.1	Регистрация и авторизация	Вход, регистрация, восстановление доступа, настройка 2FA.	12	18	15	15.33	1
2.2	Спотовая торговля	Интерфейс для покупки/продажи криптовалют, управление ордерами.	20	40	30	30	3.33
2.3	Фьючерсная торговля	Поддержка маржинальной торговли, управление ордерами с использованием плеча.	20	50	35	35	5
2.4	P2P-платформа	Разработка функционала для торговли между пользователями с поддержкой эскроу.	15	30	20	21	2.5
2.5	Инвестиционные продукты	Функционал для стейкинга, Launchpad, сберегательных продуктов.	20	35	25	26	2.5
2.6	Модуль аналитики	Реализация графиков, технических индикаторов, таблиц цен.	10	20	15	15	1.67
2.7	Депозиты и вывод средств	Интеграция платежных шлюзов, поддержка криптовалютных кошельков.	15	25	20	20.33	1.67

2.8	API для интеграции	Создание публичного и приватного API для мобильных приложений и сторонних сервисов.	10	15	12	12.33	0.83
2.9	Мобильная адаптация интерфейса	Оптимизация интерфейсов под мобильные устройства.	15	25	20	20	1.67
2.10	Программы лояльности	Разработка реферальной системы и механики скидок на комиссии.	10	20	15	15	1.67
2.11	Функционал безопасности	Реализация мониторинга подозрительных действий, защиты аккаунта, 2FA.	20	40	30	30	3.33
2.12	Образовательные материалы	Создание раздела с интерактивными курсами, гайдами и видео.	10	20	15	15	1.67
<b>3</b>	<b>Тестирование</b>						
3.1	Функциональное тестирование	Проверка корректности работы всех модулей.	15	25	20	20.33	1.67
3.2	Интеграционное тестирование	Проверка взаимодействия модулей и внешних сервисов.	12	18	15	15.33	1

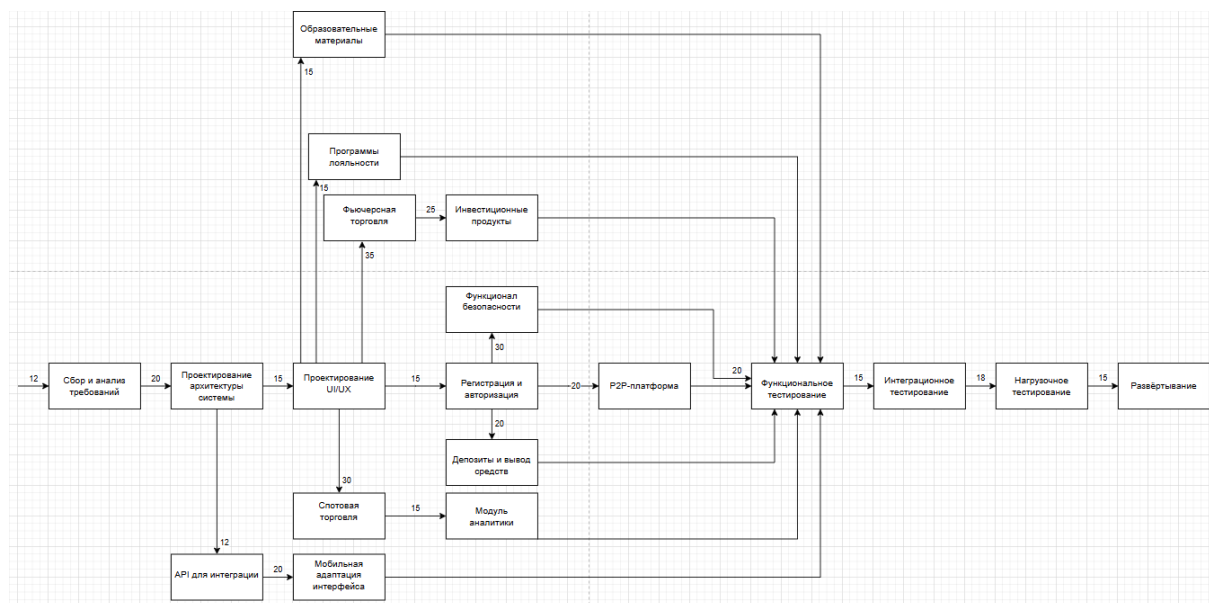
3.3	Нагрузочное тестирование	Оценка производительности под высокой нагрузкой.	12	25	18	18	2.17
<b>4</b>	<b>Внедрение</b>						
4.1	Развертывание	Установка и настройка на продакшн сервере.	10	20	15	15	1.67

Среднее общее время ( $E_i$ ): ~ **386.2 часа**.

Стандартное отклонение ( $CKO_i$ ): ~ **5.73**.

$E_{95\%} = E + 2 \times CKO = 386.2 + 2 \times 5.73 = 397.66$  часа

### 3.2 Сетевая диаграмма взаимосвязи работ



### 3.3 Метод критического пути

**Критический путь:** 141 ч./ч.

**Длинный путь:** 335.32 ч./ч.

#### Выполнение проекта:

Если ориентироваться на минимальное время выполнения критического пути, то на реализацию проекта потребуется 200 часов.

## Команда:

**Тимлид (архитектор и DevOps)** – отвечает за проектирование системы, развертывание инфраструктуры и распределение задач между участниками команды.

### 2 аналитика:

- Один специализируется на бэкенде, API и сопутствующей логике.
- Второй фокусируется на образовательных материалах и их интеграции в проект.

**2 бэкенд-разработчика** – из-за большого объема информации один разработчик физически не справится с поставленными задачами. Хотя теоретически работу может выполнять один человек, на практике это неэффективно.

**2 дизайнера** – необходимы для параллельной разработки мобильного приложения и веб-сайта. Макеты должны быть подготовлены одновременно для обеих платформ.

### 2 фронтенд-разработчика:

- Один занимается разработкой мобильного интерфейса.
- Второй отвечает за веб-интерфейс.

**1 специалист по безопасности** – проводит тестирование на уязвимости для обеспечения защиты системы от потенциальных угроз.

### 1 тестировщик

## Распределение времени по ролям:

- **Frontend:** 36 ч./ч. (5 рабочих дней)
- **Backend:** 55 ч./ч. (7 рабочих дней)
- **Тестировщик:** 20 ч./ч. (3 рабочих дня)
- **UI/UX дизайнер:** 15 ч./ч. (2 рабочих дней)
- **Аналитик:** 10 ч./ч. (2 рабочих дня)
- **Релиз и настройка:** 5 ч./ч. (1 рабочих дня)

## Рабочий процесс:

Возьмем за основу 1 рабочий день в 8 часов (с учетом перерыва).

## Параллельное выполнение задач:

Backend и Frontend можно разрабатывать параллельно, что позволит сократить общую продолжительность.

## 4. Оценка размера проекта методом функциональных точек и расчет трудоёмкости методом COSOMO II

### 4.1 Метод функциональных точек

#### Определение типа оценки:

Продукт оценивается по объему функций, необходимых для реализации криптовалютной платформы, включая управление аккаунтами, обработку транзакций, аналитические панели и т.д.

#### Определение области оценки и границ продукта:

Оцениваются все функции системы, включая основные и дополнительные. Границы системы определены на диаграммах вариантов использования (Use Case) и охватывают функциональные требования для взаимодействия с пользователями и внешними сервисами.

#### Подсчет функциональных точек, связанных с данными:

- **DET (Data Element Type)** — уникальные поля данных:
  - Например, ID пользователя, email, сумма транзакции.
- **RET (Record Element Type)** — логически сгруппированные данные:
  - Например, информация о транзакциях, личные данные пользователя.

№	Название задачи	RET	DET	Сложность	UFP
1	Регистрация пользователя	Личная информация	Имя, фамилия, email, пароль	low	3
2	Авторизация пользователя	Личная информация	Email, пароль	low	3



3	Отображение списка торговых пар	Список торговых пар	Название пары, текущая цена, объем	low	4
4	Отображение графика цен	Исторические данные по цене	Дата, время, цена открытия, закрытия	average	5
5	Создание ордера (лимит, маркет)	Ордера	Тип ордера, сумма, цена, ID пользователя	average	6
6	Просмотр истории транзакций	История транзакций	ID транзакции, сумма, дата, статус	high	7

Для оценки задач по методу функциональных точек, разложим задачи на транзакции и компоненты, определим их сложность и подсчитаем необработанные функциональные точки (UFP) и скорректированные (AFP). Определим также значение VAF на основе факторов влияния.

Каждая задача будет определена с помощью типов транзакций:

- EI (external inputs) — внешние входные транзакции (например, формы ввода).
- EO (external outputs) — внешние выходные транзакции (например, страницы с выводом данных).
- EQ (external inquiries) — внешние запросы (например, поиск).
- DET (data element type) — уникальные поля данных
- FTR (file type referenced) — количество информационных объектов, с которыми взаимодействует транзакция.

#### **Сложность:**

- *Low* — задачи с  $\leq 5$  DET и  $\leq 1$  FTR.
- *Average* — задачи с 6-10 DET или 2-3 FTR.
- *High* — задачи с  $> 10$  DET или  $> 3$  FTR.

№	Название задачи	Тип	FTR	DET	Сложность	UFP
1	Форма «Создать аккаунт»	EI	1	4	low	3
2	Авторизация пользователя	EI	1	2	low	3
3	Отображение списка активных торговых пар	EO	1	3	low	4
4	Поиск торговой пары	EQ	1	2	low	3
5	Отображение графика цен	EO	1	6	average	5
6	Создание рыночного ордера	EI	1	5	low	4
7	Создание лимитного ордера	EI	1	6	average	5
8	Отображение истории транзакций	EO	2	7	high	7
9	Удаление созданного ордера	EI	1	2	low	3
10	Настройка уведомлений по торговым сигналам	EQ	1	3	low	3
11	Отображение баланса пользователя	EO	1	4	low	4

12	Генерация отчета по финансовым операциям	EO	2	8	average	5
13	Отображение информации о комиссии при торговле	EO	1	3	low	4
14	Фильтр ордеров по типу, статусу и времени	EQ	1	3	low	3
15	Подтверждение вывода средств	EI	1	3	low	3
16	Отображение текущих рыночных условий	EO	2	7	high	7
17	Настройка двухфакторной аутентификации	EI	1	4	low	3
18	Вывод уведомлений о рыночных изменениях	EQ	1	3	low	3
19	Показ информации о торговых парах при выборе категории	EQ	1	4	low	3
20	Создание API ключа для интеграции сторонних приложений	EI	1	5	average	4
21	Переход на страницу технической поддержки	EO	1	2	low	4

22	Отображение контактной информации службы поддержки	EQ	1	2	low	3
----	--	----	---	---	-----	---

#### Подсчет суммарного количества UFP (не выровненных функциональных точек):

Для подсчета суммарного количества UFP необходимо сложить все UFP, указанные в таблице.

Сложим все значения UFP из таблицы:

$$UFP=3+3+4+3+5+4+5+7+3+3+4+5+4+3+3+7+3+3+3+4+4+3 = 86$$

#### Определение значения VAF (фактора выравнивания)

Помимо функциональных требований на продукт накладываются общесистемные требования, которые ограничивают разработчиков в выборе решения и увеличивают сложность разработки. Для учета этой сложности применяется фактор выравнивания (VAF). Значение фактора VAF зависит от 14 параметров, которые определяют системные характеристики продукта:

№	Параметр	Описание	Оценка параметра (DI)
1	Обмен данными	Наличие функциональности для передачи и синхронизации данных между системами или модулями.	3
2	Распределенная обработка данных	Сложность в обработке данных, выполняемой на нескольких серверах или устройствах.	2

3	Производительность	Требования к быстродействию системы, включая скорость обработки запросов и загрузки данных.	4
4	Ограничения по аппаратным ресурсам	Учет требований к минимальной конфигурации оборудования, на котором должна работать система.	2
5	Транзакционная нагрузка	Количество транзакций, выполняемых в единицу времени, и их сложность.	3
6	Интенсивность взаимодействия с пользователем	Частота операций, выполняемых пользователем, включая элементы интерфейса, которые требуют взаимодействия.	4
7	Эргономика	Уровень удобства использования системы для конечного пользователя, включая доступность и интуитивность интерфейса.	3
8	Сложность обработки	Обработка данных с учетом сложных алгоритмов, многошаговых процессов или зависимости от внешних источников.	5
9	Повторное использование	Возможность повторного использования компонентов системы в других проектах или модулях.	2
10	Удобство инсталляции	Простота установки программного обеспечения, включая	3

		автоматизированные процессы и инструкции для пользователя.	
11	Удобство администрирования	Удобство управления и настройки системы, включая мониторинг состояния, доступ к журналам событий и интерфейсы для управления пользователями.	4
12	Портируемость	Возможность адаптации системы для работы на других платформах или устройствах без существенных изменений.	2
13	Гибкость	Уровень адаптируемости системы к изменению функциональных требований или добавлению новых функций.	3
14	Масштабируемость	Возможность системы работать при увеличении нагрузки, например, при росте числа пользователей или объемов данных.	3

### Расчет суммарного эффекта:

#### Суммарный эффект 14 системных характеристик (TDI):

$$TDI = \sum DI = 43$$

#### Расчет фактора выравнивания (VAF):

##### Формула:

$$VAF = (TDI \times 0.01) + 0.65 = (43 \times 0.01) + 0.65 = 1.08 = 1.08$$

### Подсчет количества AFP (выровненных функциональных точек)

$$AFP = UFP * VAF = 86 \times 1.08 = 92.88$$

Метод анализа функциональных точек не предоставляет информации о трудоемкости разработки оцениваемого продукта. Этот вопрос легко решается, если у компании-разработчика есть собственная статистика трудозатрат на реализацию функциональных точек. В случае отсутствия такой статистики, для оценки трудоемкости и сроков выполнения проекта можно воспользоваться методом COCOMO II.

## 4.2 Метод COCOMO II

### Оценка размера продукта в KSLOC

Главной особенностью методики является то, что для того, чтобы оценить трудоемкость, необходимо знать размер программного продукта в тысячах строках исходного кода (KSLOC, Kilo Source Lines Of Code).

<https://www.qsm.com/resources/function-point-languages-table>

### Технологический стек:

#### Frontend:

- React.js

#### Backend:

- PostgreSQL

### Расчеты:

#### 1. Frontend:

$$KSLOC_{front} = 135 \times (43 \times 0.047) = 135 \times 0.03525 = 4.75$$

#### 2. Backend:

$$KSLOC_{back} = 135 \times (41 \times 0.037) = 135 \times 0.00925 = 1.25$$

#### 3. Общий размер продукта:

$$KSLOC = KSLOC_{front} + KSLOC_{back} = 4.75 + 1.25 = 6.0$$

## Факторы масштаба

В методике используются пять факторов масштаба SF, которые определяются следующими характеристиками проекта:

1. PREC — прецедентность, наличие опыт аналогичных разработок **NOMINAL**
2. FLEX — гибкость процесса разработки **HIGH**
3. RESL — архитектура и разрешение рисков **NOMINAL**
4. TEAM — сработанность команды **HIGH**
5. PMAT — зрелость процессов **NOMINAL**

## Оценка уровня факторов масштаба:

№	Факторы масштаба	Оценка уровня фактора
1	PREC	3.72
2	FLEX	2.03
3	RESL	4.24
4	TEAM	2.19
5	PMAT	4.68

## Множители трудоемкости

Рассмотрим случай предварительной оценки трудоемкости программного проекта. Для этой оценки необходимо оценить для проекта уровень семи множителей трудоемкости M<sub>i</sub>:-

1. PERS — квалификация персонала **HIGH**
2. RCPX — сложность и надежность продукта **NOMINAL**
3. RUSE — разработка для повторного использования **LOW**
4. PDIF — сложность платформы разработки **LOW**
5. PREX — опыт персонала **NOMINAL**
6. FCIL — оборудование **NOMINAL**
7. SCED — сжатие расписания **NOMINAL**



### Оценка уровня множителей трудоемкости:

№	Множители трудоемкости	Оценка уровня множителя
1	PERS	0.83
2	RCPX	1.00
3	RUSE	0.95
4	PDIF	0.87
5	PREX	1.00
6	FCIL	1.00
7	SCED	1.00

### Оценка многокомпонентного продукта

Для оценки трудоемкости проекта в контексте многокомпонентной разработки по методике COCOMO II используется следующая формула:

$$E = 0.91 + 0.01 \times \sum SF_j$$

где:

- SIZE — размер продукта в KSLOC (тысячах строк исходного кода)
- E<sub>Mi</sub> — множители трудоемкости
- SF<sub>j</sub> — факторы масштаба
- n = 7 — для предварительной оценки
- n = 17 — для детальной оценки

$$E = 0.91 + 0.01 \times (3.72 + 2.03 + 4.25 + 2.19 + 4.68) = 1,0786$$

#### 1. Суммарный размер продукта в KSLOC:

$$SIZE^A = \sum_{k=1}^N SIZE_k$$

SIZE = 6 (суммарный размер компонентов в KSLOC)

2. **Базовая трудоемкость проекта (PMB):**

$$PM^B = A \times (SIZE^A)^E \times SCED$$

$$PM^B = 2.94 \times 6.574 = 19.36 \text{ чел.} \cdot \text{мес.}$$

3. **Базовая трудоемкость для каждого компонента (PMB<sub>k</sub>):**

$$PM_k^B = PM^B \times \frac{SIZE_k}{SIZE^A}$$

Для многокомпонентного проекта можно вычислить трудоемкость каждого компонента. Поскольку у нас только один компонент, базовая трудоемкость для компонента будет такой же:

$$PM_k^B = 19.36$$

4. **Оценка трудоемкости компонента с учетом множителей трудоемкости (кроме множителя SCED):**

Мы учитываем все множители трудоемкости, кроме множителя SCED, для получения более точной оценки:

$$PM'_k = PM_k^B \times \prod_{i=1}^6 EM_i$$

$$PM'_k = PMB \times (1.0786 \times 0.83 \times 1.0786 \times 0.95 \times 1.0786 \times 0.87) = 16.61 \text{ чел.} \cdot \text{мес}$$

5. **Итоговая трудоемкость проекта (PM):**

$$PM = \sum_{k=1}^N PM'_k$$

$$PM = 16.61 \text{ чел.} \cdot \text{мес} \approx 265.76 \text{ ч.ч.}$$

## 5. Оценка размера методом Use Case Points



### Оценка веса прецедентов

Сложность	Вес (UUCW)	Количество	Затраты
LOW	5	7	35
MID	10	8	80
HIGH	15	3	45
Нескорректированный вес варианта использования (UUCW)			160

### Оценка веса акторов

Сложность	Вес (AWi)	Количество	Затраты
LOW	1	2	2
MID	2	4	8
HIGH	3	2	6
Масса акторов без корректировки (UAW)			16

### Оценка веса технических факторов

Фактор	Описание	Вес	Сложность	Затраты
T1	Распределенная система	2	3	6
T2	Время отклика	1	2	2
T3	Эффективность для конечного пользователя	1	2	2
T4	Сложность внутренней разработки	2	2	4
T5	Возможность повторного использования кода	1	3	3
T6	Простота установки	0.5	1	0.5
T7	Простота в использовании	0.5	2	1
T8	Переносимость на другие платформы	1	3	3
T9	Техническое обслуживание системы	1	2	2
T10	Параллельная обработка	1	1	1

T11	Функции безопасности	2	3	6
T12	Доступ для третьих лиц	1	1	1
T13	Обучение конечных пользователей	1	0	0
Общий технический фактор (TFactor)				31.5
TCF = 0,6 + (TF/100) =				0,915

Фактор	Описание	Вес	Влияние	Затраты
E1	Знакомство с используемым процессом разработки	1.5	4	6
E2	Опыт применения технологий	0.5	3	1.5
E3	Объекто-ориентированный опыт работы команды	1	3	3
E4	Возможности ведущего аналитика	0.5	2	1
E5	Мотивация команды	1	2	2
E6	Стабильность требований	2	2	4
E7	Сотрудники, занятые неполный рабочий день	-1	3	-3
E8	Сложный язык программирования	-1	3	-3
Общий фактор окружающей среды (EFactor)				11.5
EFC = 1.4 + (-0.03 × EFactor)				1.055

$$UCP = (UUCW + UAW) \times TCF \times ECF = (160 + 16) \times 0.915 \times 1.055 = 310 \times 0.83 \times 1.055 = 170$$

### Подсчет фактора продуктивности на основе прошлого проекта

В качестве примера выбрали проект по предмету «Технологии разработки интерактивных цифровых ресурсов», который был разработан нами – командой из двух студентов-разработчиков.

## Список UseCase'ов

№	Сценарий
1	Ввод id аккаунта
2	Просмотр истории матчей
3	Просмотр лучших команд
4	Мини-игра Invoker Game
5	Таблица лидеров Invoker Game
6	Личные рекорды
7	Вход при помощи Steam API

## Оценка веса прецедентов

Сложность	Вес (UUCW)	Количество	Затраты
LOW	5	2	10
MID	10	4	40
HIGH	15	1	15
Нескорректированный вес варианта использования (UUCW)			65

## Оценка веса акторов

Сложность	Вес (AWi)	Количество	Затраты
LOW	1	0	0
MID	2	2	4
HIGH	3	1	3
Масса акторов без корректировки (UAW)			7

### Оценка веса технических факторов

Фактор	Описание	Вес	Сложность	Затраты
T1	Распределенная система	2	3	6
T2	Время отклика	1	3	3
T3	Эффективность для конечного пользователя	1	3	3
T4	Сложность внутренней разработки	1	3	3
T5	Возможность повторного использования кода	1	2	2
T6	Простота установки	0.5	2	1
T7	Простота в использовании	0.5	3	1.5
T8	Переносимость на другие платформы	2	1	2
T9	Техническое обслуживание системы	1	2	2
T10	Параллельная обработка	1	0	0
T11	Функции безопасности	1	3	3
T12	Доступ для третьих лиц	1	0	0
T13	Обучение конечных пользователей	1	0	0
Общий технический фактор (TFactor)				26.5
$TCF = 0,6 + (TF/100) =$				0,83

### Общий фактор окружения

Фактор	Описание	Вес	Влияние	Затраты
E1	Знакомство с используемым процессом разработки	1.5	4	6
E2	Опыт применения технологий	0.5	3	1.5
E3	Объекто-ориентированный опыт работы команды	1	3	3
E4	Возможности ведущего аналитика	0.5	2	1

E5	Мотивация команды	1	2	2
E6	Стабильность требований	2	3	6
E7	Сотрудники, занятые неполный рабочий день	-1	3	-3
E8	Сложный язык программирования	-1	3	-3
Общий фактор окружающей среды (EFactor)				13.5
$EFC = 1.4 + (-0.03 \times EFactor)$				1.055

$$UCP = (UUCW + UAW) \times TCF \times ECF = (65 + 7) \times 0.83 \times 1.055 = 310 \times 0.83 \times 1.055 = 60$$

### Подсчет трудоемкости проекта:

Данный проект был выполнен за 90 часов (работали 2 человека).

$$PF = 90 / 60 = 1$$

UCP анализируемого веб-проекта - 170. Следовательно,

$$E = PF * UCP = 1.5 * 170 = 255 \text{ ч/ч}$$

## 6. Сравнение результатов и выводы

Метод	Затраты (ч./ч.)
Наивный метод	452,9
PERT	397,66
COCOMO II	266
UCP	255

### Анализ методов:

#### 1. Наивный метод (452,9 ч./ч.):

Наивный метод основан на предположениях и опыте, что делает его быстрым, но менее точным. Он выдал самую низкую оценку, так как не учитывает сложность и уникальные особенности проекта. Этот метод подходит для начальной грубой оценки, но не для детального планирования.



2. **PERT (234,31 ч./ч.):**

Метод PERT использует три оценки (оптимистическую, пессимистическую и наиболее вероятную), что делает его более точным, чем наивный метод. Полученная оценка близка к результатам UCP, что говорит о ее объективности. Этот метод хорошо подходит для проектов средней сложности.

3. **COCOMO II (266 ч./ч.):**

COCOMO II выдал наивысшую оценку трудоемкости. Этот метод учитывает множество факторов, включая сложность проекта, масштаб и среду разработки. Однако в данном случае COCOMO II, вероятно, завысил трудоемкость, так как текущий проект (например, сайт театра или небольшой веб-проект) не требует такого уровня сложности, как крупные корпоративные системы.

4. **UCP (255 ч./ч.):**

Метод UCP (Use Case Points) ориентирован на функциональные требования и учитывает количество и сложность вариантов использования. Он показал реалистичную оценку, которая согласуется с PERT. Это делает UCP подходящим выбором для оценки веб-проектов средней сложности.

**Вывод:**

Наиболее реалистичными методами для данного проекта являются **PERT** и **UCP**. Эти методы учитывают функциональные и вероятностные аспекты, что особенно важно для веб-разработки.

**COCOMO II** ориентирован на проекты с высокой сложностью и крупным масштабом, что не соответствует текущему проекту, поэтому выдал результат больше чем PERT и UCP.

**Наивный метод**, напротив, переоценивает трудоемкость, так как не учитывает специфику разработки и сложность системы.