Министерства науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**Лабораторная работа №1**

По дисциплине

«Экономика программной инженерии»

Студент:

Иванов Артемий Антонович P34232

Преподаватель:

Машина Екатерина Алексеевна



**Санкт-Петербург**

**2024**

Оглавление

[**Задание 3**](#_padjpxnl4912)

[**Функциональные требования 3**](#_g22py6rcowtv)

[1. Система должна предоставлять регистрацию и авторизацию пользователей 3](#_nj3up9zd2wqw)

[2. Система должна предоставлять работу с задачами 3](#_sy6d4gus1zhs)

[3. Система должна предоставлять интерактивную карту 4](#_qiplhurstyz3)

[5. Система должна предоставлять систему рейтингов и достижений 4](#_hppgvxiyrod9)

[7. Система должна предоставлять дополнительные функции 4](#_3wz9nfu76kxr)

[**Оценка трудоемкости наивным методом 5**](#_bnzybb31a4m4)

[**Оценка трудоемкости методом PERT 6**](#)

[**Сетевая диаграмма взаимосвязи работ 7**](#_gjvgjs6rav9k)

[**Расчет минимальной продолжительности проекта методом критического пути 7**](#_n1zcs8vvxp61)

[**Метод функциональных точек 8**](#_e8uis595q5b8)

[1. Определение функциональных компонентов 8](#_ux7nxyucs4yy)

[a. Внешние входы (EI) 8](#_f4fn84gh3dfr)

[b. Внешние выходы (EO) 8](#_91o4nk3xyh4y)

[c. Внешние запросы (EQ) 8](#_2lny91572841)

[d. Внутренние логические файлы (ILF) 8](#_3b0lgwnabwr9)

[e. Внешние интерфейсные файлы (EIF) 8](#_9o7symunc7h2)

[2. Взвешивание функциональных компонентов 8](#_ykiru9m0efuy)

[3. Коррекция по факторам сложности 9](#_r86vaeeozof7)

[Итог 9](#_6bspmlhii9xo)

[**Расчет трудоемкости методом COCOMO II 9**](#_77k5yi67cjzc)

[Определение факторов масштаба (Scale Factors, SF) 9](#_ismlcn311ogt)

[Формула времени разработки (TDEV): 12](#_dcenm6xu11x4)

[**Use Case Points 13**](#_x48wuvr5wp5b)

[Определение параметров UCP 13](#_o15mi86x4lh6)

[a. Выделение акторов 13](#_aw90n14gr80)

[b. Выделение сценариев использования (use cases) 13](#_uut50n8rwfhy)

[Коррекция на техническую сложность (TCF) 14](#_ku55nb2ytpgk)

[a. Определение технических факторов 14](#_jqhdf8iywxqm)

[Коррекция на фактор среды (EF) 15](#_z7zo3mdjlq6b)

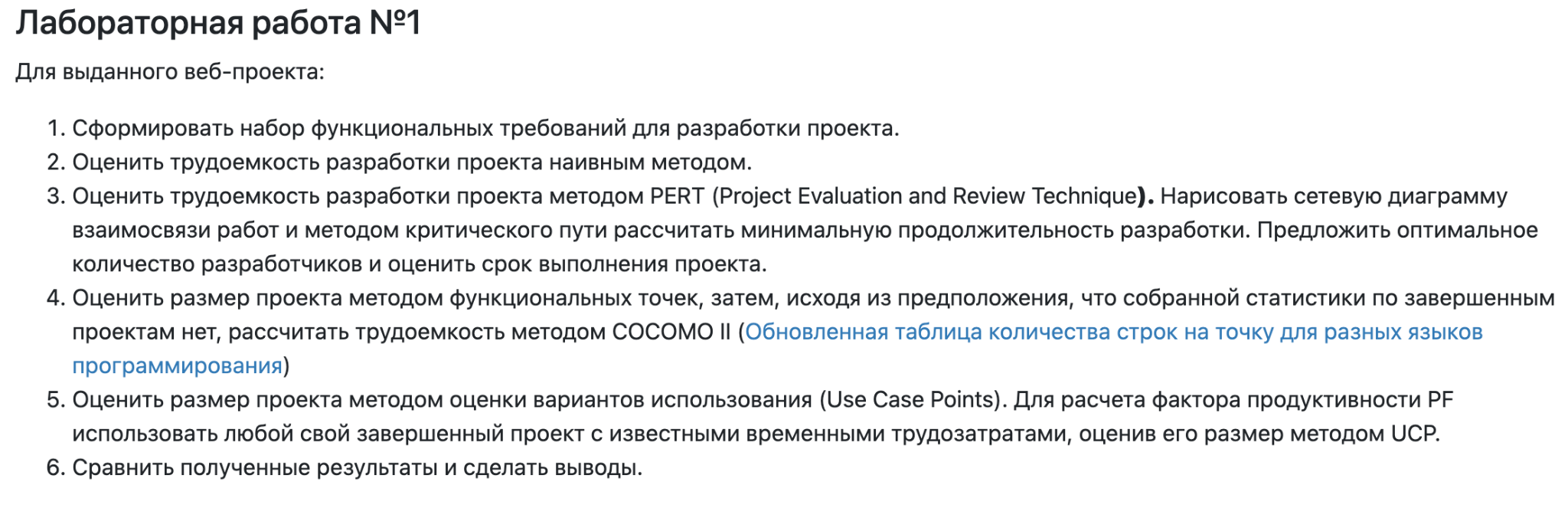
[a. Определение факторов среды 15](#_l4cwu5xg3wta)

[Расчет итоговых use case points (UCP) 15](#_1w6n3192sewj)

[Итог 16](#_hzv14dz2aj7v)

[**Заключение 16**](#_ttzps2ur2sx)

# Задание



Вариант: [https://maproulette.org](https://maproulette.org/)

# Функциональные требования

#### **1. Система должна предоставлять регистрацию и авторизацию пользователей**

* Возможность регистрации через электронную почту и пароль.
* Авторизация через социальные сети (Google, Facebook).
* Сброс и восстановление пароля.
* Профиль пользователя, включающий:
  + Имя.
  + Аватар.
  + Информация о прогрессе (количество выполненных задач, достижения).

#### **2. Система должна предоставлять работу с задачами**

* Возможность пользователям создавать задачи:
  + Указание категории задачи (например, картографическая ошибка, проверка данных и т.д.).
  + Указание географического региона (на карте).
  + Добавление описания задачи, инструкции и вложений (скриншоты, ссылки).
* Список задач с фильтрацией:
  + По категории.
  + По уровню сложности (например, легкий, средний, сложный).
  + По географическому положению (в радиусе, по конкретной области).
  + По статусу (выполнено, в процессе, доступно).
* Прогресс выполнения задач:
  + Визуализация прогресса (процент выполненных задач).
  + История выполнения задачи (кто, когда и что делал).

#### **3. Система должна предоставлять интерактивную карту**

* Карта для отображения задач:
  + Задачи отображаются как маркеры или полигоны.
  + При клике на задачу — открывается информация о ней.
* Возможность фильтрации задач на карте.
* Отображение выполненных и активных задач разными цветами.
* Возможность масштабирования и перемещения по карте.

**4. Система должна предоставлять механизм выполнения задач**

* Интерфейс для выполнения задачи:
  + Поле для ввода текста (комментарии или решения).
  + Кнопки для отметки задачи как "выполнено", "невозможно выполнить", "передано на доработку".
* Подсказки или инструкции для каждой задачи.
* Возможность загрузки файлов, связанных с выполнением задачи.
* Встроенная система комментариев для обсуждения задачи.

#### **5. Система должна предоставлять систему рейтингов и достижений**

* Начисление баллов за выполнение задач.
* Лидеры недели/месяца/всего времени.
* Значки и уровни за выполнение определенного числа задач или достижение целей.

**6. Система должна предоставлять систему общения между пользователями**

* Возможность обсуждения задач в чате или форуме.
* Подписка на задачи или группы задач.
* Уведомления:
  + О новых задачах.
  + О комментариях и прогрессе по вашим задачам.
* Возможность работать над задачами в командах:
  + Создание команд.
  + Рейтинг и прогресс команд.

#### **7. Система должна предоставлять дополнительные функции**

* Режим обучения для новых пользователей.
* Экспорт данных о задачах в CSV/JSON.
* Интеграция с внешними сервисами (JOSM для редактирования данных карты).

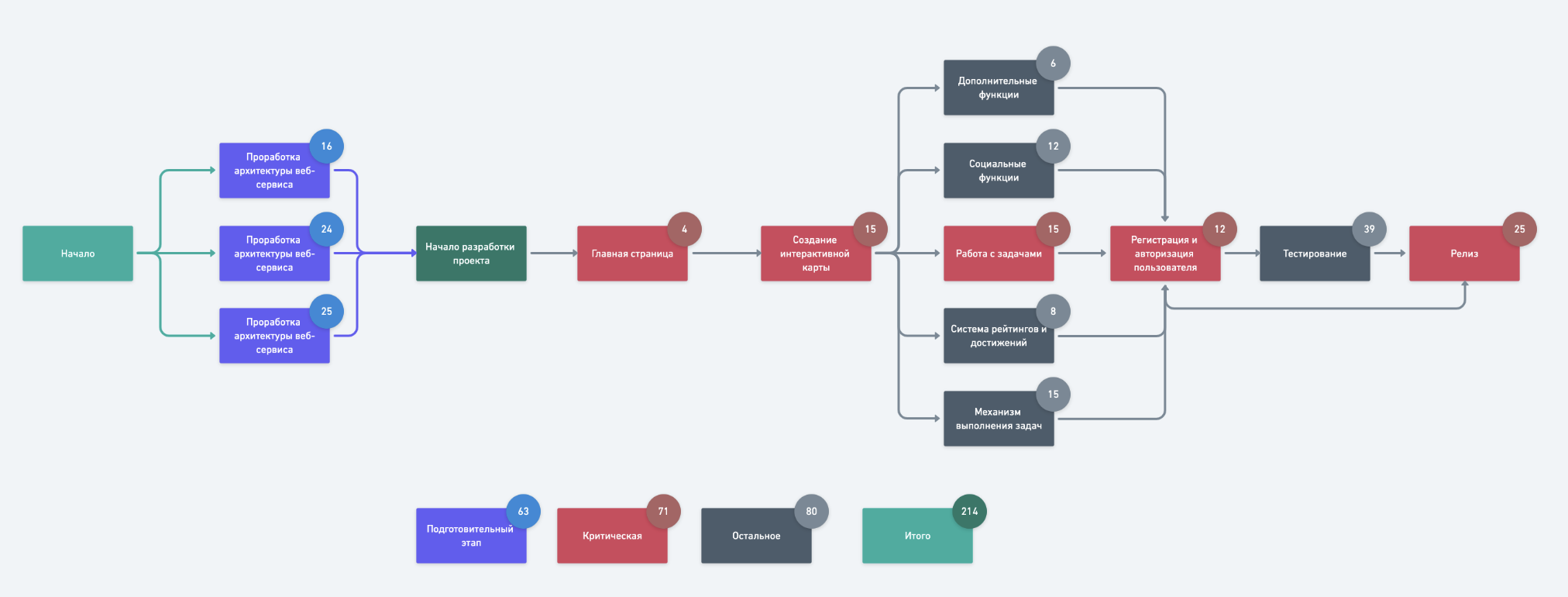
# Оценка трудоемкости наивным методом

| **№** | **Функционал** | **Optimistic** | **Optimal** | **Pessimistic** |
| --- | --- | --- | --- | --- |
| 1 | Регистрация и авторизация пользователя | 8 | 12 | 16 |
| 2 | Работа с задачами | 20 | 28 | 35 |
| 3 | Интерактивная карта | 10 | 15 | 20 |
| 4 | Механизм выполнения задач | 15 | 18 | 21 |
| 5 | Система рейтингов и достижений | 6 | 8 | 11 |
| 6 | Социальные функции | 10 | 12 | 14 |
| 7 | Дополнительные функции | 4 | 6 | 7 |
| 8 | Тестирование | 25 | 40 | 50 |
| 9 | Релиз | 15 | 25 | 40 |
| **∑** | | **113** | **164** | **214** |

# Оценка трудоемкости методом PERT

| **№** | **Функционал** | **Optimistic** | **Optimal** | **Pessimistic** | **Ei = (Pi + 4Mi + Oi)/6** | **CKOi = (Pi**  **- Oi)/6.** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Регистрация и авторизация пользователя | 8 | 12 | 16 | 12 | 4 |
| 2 | Работа с задачами | 20 | 28 | 35 | 18 | 6 |
| 3 | Интерактивная карта | 10 | 15 | 20 | 15 | 5 |
| 4 | Механизм выполнения задач | 15 | 18 | 21 | 18 | 6 |
| 5 | Система рейтингов и достижений | 6 | 8 | 11 | 8.166666667 | 2.833333333 |
| 6 | Социальные функции | 10 | 12 | 14 | 12 | 4 |
| 7 | Дополнительные функции | 4 | 6 | 7 | 5.833333333 | 1.833333333 |
| 8 | Тестирование | 25 | 40 | 50 | 39.16666667 | 4.166666667 |
| 9 | Релиз | 15 | 25 | 40 | 25.83333333 | 4.166666667 |
| **∑** | | **113** | **164** | **214** | **154** | **38** |

# Сетевая диаграмма взаимосвязи работ



# Расчет минимальной продолжительности проекта методом критического пути

Критический путь 71 ч

| Минимальная продолжитель ность разработки (методом критического пути) / час | Суммарная трудоемкость проекта / чел.час E9 | Количество рабочих часов в день на одного разработчика / час | Срок выполнения проекта / рабочие дни | Количество  разработчиков  / чел |
| --- | --- | --- | --- | --- |
| 71 | 214 | 8 | 9 | 3 |

# 

# Метод функциональных точек

### **1. Определение функциональных компонентов**

#### **a. Внешние входы (EI)**

* Авторизация пользователя: 1 EI
* Создание/редактирование вызовов: 2 EI
* Внесение изменений в задачи (отметка завершения): 1 EI
* Поиск задач: 1 EI  
  **Итого: 5 EI**

#### **b. Внешние выходы (EO)**

* Отображение списка задач: 1 EO
* Вывод аналитики и статистики: 1 EO  
  **Итого: 2 EO**

#### **c. Внешние запросы (EQ)**

* Получение данных задач: 1 EQ
* Получение деталей задачи: 1 EQ
* Получение статистики: 1 EQ  
  **Итого: 3 EQ**

#### **d. Внутренние логические файлы (ILF)**

* База данных задач: 1 ILF
* База пользователей: 1 ILF  
  **Итого: 2 ILF**

#### **e. Внешние интерфейсные файлы (EIF)**

* Интерфейс с OpenStreetMap API: 1 EIF
* Интерфейс с аналитической системой: 1 EIF  
  **Итого: 2 EIF**

### **2. Взвешивание функциональных компонентов**

Теперь взвесим компоненты, используя таблицу весов:

| **Тип компонента** | **Легкий** | **Средний** | **Сложный** |
| --- | --- | --- | --- |
| **EI** | 3 FP | 4 FP | 6 FP |
| **EO** | 4 FP | 5 FP | 7 FP |
| **EQ** | 3 FP | 4 FP | 6 FP |
| **ILF** | 7 FP | 10 FP | 15 FP |
| **EIF** | 5 FP | 7 FP | 10 FP |

Мы предположим средний уровень сложности для всех компонентов.

* **EI**: 5×4=20
* **EO**: 2×5=10
* **EQ**: 3×4=12
* **ILF**: 2×10=20
* **EIF**: 2×7=14

**Итого без учета факторов сложности: 76 функциональных точек.**

### **3. Коррекция по факторам сложности**

Корректируем итоговое значение с помощью 14 факторов влияния (Value Adjustment Factor, VAF), таких как производительность, доступность, надежность и т.д. Предположим среднее значение веса: VAF=1.1 × VAF = 1.1 ×VAF=1.1.

**Общее количество функциональных точек:**FP=76×1.1=83.6FP = 76 × 1.1 = 83.6FP=76 × 1.1=83.6

### **Итог**

**84 функциональные точки**.

# Расчет трудоемкости методом COCOMO II

### **Определение факторов масштаба (Scale Factors, SF)**

Факторы масштаба:

| **Фактор** | **Оценка** | **Значение** |
| --- | --- | --- |
| PREC | Номинальная | 3,72 |
| FLEX | Номинальная | 3,04 |
| RESL | Номинальная | 4,24 |
| TEAM | Номинальная | 3,29 |
| PMAT | Номинальная | 4,68 |
| **Итого** | Номинальная | **18,97** |

Формула для вычисления коэффициента масштаба:

SF=∑SFi=18,97

**Определение множителей трудоемкости (Effort Multipliers, EM)**

Используем предложенные множители:

| **Фактор** | **Оценка** | **Значение** |
| --- | --- | --- |
| PERS | Номинальная | 1,00 |
| RCPX | Номинальная | 1,00 |
| RUSE | Номинальная | 1,00 |
| PDIF | Номинальная | 1,00 |
| PREX | Номинальная | 1,00 |
| FCIL | Номинальная | 1,00 |
| SCED | Номинальная | 1,00 |

Произведение множителей:

∏EMi=1,00

**Коэффициент конверсии для языка Python**

Размер проекта в функциональных точках — 84 (из предыдущего расчета). Используем коэффициент конверсии для Python (939393 строк на функциональную точку):

KLOC=84×93/1000=7,812 KLOC

**Базовая формула для трудоемкости (PM)**

Формула трудоемкости:

E=A×SIZEB+0.01×SF×∏EMi

Где:

* A=2,94 (константа для COCOMO II)
* B=0,91 (базовая экспонента)
* SF=18,97 (сумма факторов масштаба)
* ∏EMi=1,00 (произведение множителей трудоемкости).

Подставляем значения:

E=2,94×7,8120,91+0,01×18,97×1,00

Вычислим значение экспоненты:

Bfinal=0,91+0,01×18,97=1,0997

Теперь рассчитаем трудоемкость:

E=2,94×7,8121,0997=2,94×9,765=28,71 человеко-месяцев

**Время разработки и распределение усилий**

#### **Формула времени разработки (TDEV):**

TDEV=C×EDTDEV

Где:

* C=3,67
* D=0,28

Подставляем:

TDEV=3,67×(28,71)0,28=3,67×3,45=12,66 месяцев

**Средний размер команды:**

Команда=ETDEV=28,7112,66≈2,27 человека

**Итоговые результаты**

1. **Трудоемкость (PM):** 28,7 человеко-месяцев.
2. **Время разработки (TDEV):** 12,7 месяцев.
3. **Средний размер команды:** 2-3 человека.

# Use Case Points

### **Определение параметров UCP**

#### **a. Выделение акторов**

Акторы — это пользователи или внешние системы, которые взаимодействуют с системой. Они классифицируются по уровню сложности:

| **Тип актора** | **Пример** | **Вес** |
| --- | --- | --- |
| **Простой** (Simple) | Актор, использующий стандартный API (например, OpenStreetMap API). | 1 |
| **Средний** (Average) | Пользователь с графическим интерфейсом. | 2 |
| **Сложный** (Complex) | Сложная система или другой сайт с безопасными API. | 3 |

Для сайта **MapRoulette** можно выделить акторов:

* **Пользователь** (средний) — 1 актор, вес 2.
* **OpenStreetMap API** (простой) — 1 актор, вес 1.
* **Администратор** (сложный) — 1 актор, вес 3.

Общая оценка акторов:

UA=(1×2)+(1×1)+(1×3)=6

#### **b. Выделение сценариев использования (use cases)**

| **Тип сценария** | **Пример** | **Вес** |
| --- | --- | --- |
| **Простой** (Simple) | До 3 шагов. | 5 |
| **Средний** (Average) | От 4 до 7 шагов. | 10 |
| **Сложный** (Complex) | Более 7 шагов. | 15 |

Сценарии:

1. **Авторизация пользователя** — простой (5).
2. **Просмотр списка задач** — средний (10).
3. **Выполнение задачи** — сложный (15).
4. **Создание/редактирование вызовов (админ)** — сложный (15).
5. **Получение аналитики** — средний (10).

Общая оценка сценариев:

UC=(1×5)+(1×10)+(1×15)+(1×15)+(1×10)=55.UC

**c. Вычисление необработанных use case points (UUCP)**

Формула:

UUCP=UA+UC

Подставляем значения:

UUCP=61

### **Коррекция на техническую сложность (TCF)**

#### **a. Определение технических факторов**

Оценим технические факторы, влияющие на сложность разработки. Каждому фактору дается вес и оценка от 0 до 5 (где 0 — незначительное влияние, 5 — максимальное влияние).

| **Фактор** | **Вес** | **Оценка** | **Результат** |
| --- | --- | --- | --- |
| Высокая надежность | 2 | 3 | 6 |
| Сложность интерфейса | 1 | 2 | 2 |
| Мобильный/веб-доступ | 3 | 4 | 12 |
| Сложность обработки данных | 2 | 3 | 6 |
| Интеграция с внешними системами | 3 | 4 | 12 |

Итого:

TF=∑(Вес×Оценка)=6+2+12+6+12=38

Формула для TCF:

TCF=0.6+(0.01×TF)

Подставляем:

TCF=0.6+(0.01×38)=0.98

### **Коррекция на фактор среды (EF)**

#### **a. Определение факторов среды**

| **Фактор** | **Вес** | **Оценка** | **Результат** |
| --- | --- | --- | --- |
| Опыт работы с похожими проектами | 1 | 3 | 3 |
| Опыт работы с языком (Python) | 2 | 4 | 8 |
| Знание предметной области | 2 | 3 | 6 |
| Мотивация команды | 1 | 5 | 5 |
| Инструменты разработки | 1 | 4 | 4 |

Итого:

EF=∑(Вес×Оценка)=3+8+6+5+4=26

Формула для EF:

EF=1.4−(0.03×EF)

Подставляем:

EF=1.4−(0.03×26)=0.62

### **Расчет итоговых use case points (UCP)**

Формула:

UCP=UUCP×TCF×EF

Подставляем значения:

UCP=61×0.98×0.62=37.07

**Оценка трудоемкости (PF)**

Для оценки трудоемкости используем коэффициент производительности (Productivity Factor, PF).

PF = 20

Формула:

PM=UCP×PF

Подставляем:

PM=37.07×20=741.4 часы

### **Итог**

1. **Необработанные UCP (UUCP):** 61.
2. **Технический коэффициент (TCF):** 0.98.
3. **Коэффициент среды (EF):** 0.62.
4. **Итоговые UCP:** 37.07.
5. **Трудоемкость:** ~741.4 часов (~92 человеко-дней, если брать 8 часов в день).

# Заключение

В результате исследования различных способов оценки трудозатрат было выяснено, что методы COCOMO II и Use Case Points способны выдавать более адекватные и правдоподобные результаты для проектов. Это объясняется тем, что данные методы учитывают множество факторов, способствующих изменению сложности и объема работ. Рекомендуем внедрять эти методики в планирование проекта, чтобы обеспечить достаточные ресурсы и снизить вероятность рисков, связанных с недооценкой трудозатрат.