

Санкт-Петербургский национальный исследовательский  
университет

информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники

**Лабораторная работа № 1 по дисциплине**  
**«Экономика программных систем»**

Сайт [mangalib.me](http://mangalib.me)

Выполнили:

Студенты группы Р43131

Ибадуллаев Алибаба

Осипов Василий

Преподаватель: Машина Е. А.

Санкт-Петербург

2023

# Оглавление

<b>Оглавление</b> .....	<b>2</b>
Задание.....	2
Вариант.....	3
Выполнение.....	3
Функциональные требования.....	3
Оценка трудоемкости работ (наивный метод).....	5
PERT-метод.....	6
Оценка размера проекта методом функциональных точек (FP).....	10
Расчет трудоемкости методом COCOMO II.....	13
Тип проекта.....	13
Определение размера проекта в строках кода.....	13
Расчет фактора коррекции EAF (Effort Adjustment Factor).....	13
Расчет трудоемкости проекта с использованием формулы COCOMO II.....	15
Расчет продолжительности проекта и необходимого количества человеко-месяцев.....	15
Метод оценки вариантов использования.....	15
<b>Сравнение использованных методов</b> .....	<b>21</b>
<b>Вывод</b> .....	<b>22</b>

## 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 COCOMO 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

Сайт <http://mangalib.me>

## Выполнение

### Функциональные требования

1. Система должна предоставлять возможность выбора интересующей манги, путем просмотра, фильтрации и поиска по каталогу манги и комиксов:
  - 1.1. Возможность отфильтровать мангу по жанрам, тегам, типам, формату выпуска, статусам переводов, статусам тайтлов.
  - 1.2. Возможность отсортировать мангу по рейтингу, кол-ву оценок, просмотрам, дате добавления/обновления, количеству глав в порядках возрастания/убывания для каждого параметра.
  - 1.3. Возможность поиска манги по названию.
  - 1.4. Возможность выбора случайной манги.
2. Система должна предоставлять возможность просмотра информации о конкретной манге/комиксе:
  - 2.1. Просмотр краткого описания сюжета, связанных тегов, переводчиков, автора, статуса, рейтинга и пр. информации.
  - 2.2. Просмотр связанных и похожих манг/комиксов.
  - 2.3. Просмотр комментариев и обсуждений, связанных с конкретной мангой.
  - 2.4. Просмотр списка томов и глав конкретной манги, с возможностью их открытия в режиме чтения
3. Система должна предоставлять возможность выполнять с конкретной мангой/комиксом следующие действия (помимо чтения):
  - 3.1. Возможность оценить мангу/комикс по десятибалльной шкале.
  - 3.2. Возможность оставить комментарий
  - 3.3. Возможность перейти на форум для создания обсуждения
  - 3.4. Возможность добавления тайтла в список
    - 3.4.1. Читаю

- 3.4.2. В планах
  - 3.4.3. Брошено
  - 3.4.4. Прочитано
  - 3.4.5. Любимые
  - 3.4.6. \*Свой произвольный список\*
- 3.5. Возможность отправки сообщения об ошибке модератору
- 3.6. Получение уведомлений о выходе новых глав на главной странице
- 4. Публикация контента. Все эти действия должны пройти модерацию перед их всеобщей публикацией:
  - 4.1. Возможность внесения изменений в описание тайтла
  - 4.2. Добавление отдельных глав в тайтл
  - 4.3. Возможность добавления тайтла
  - 4.4. Возможность добавления автора
  - 4.5. Возможность добавления команды
- 5. Чтение манги и комиксов:
  - 5.1. Возможность онлайн-чтения манги и комиксов.
  - 5.2. Возможность изменения настроек чтения (тема, режим чтения, сервера, размера страницы читалки)
  - 5.3. Поддержка пагинации (перелистывание страниц, томов и глав) и масштабирования изображений.
  - 5.4. Возможность комментирования отдельных страниц
  - 5.5. Возможность оценивания комментариев пользователей
  - 5.6. Возможность просмотра манги со страницы, на которой оставлена закладка
- 6. Учетные записи пользователей:
  - 6.1. Регистрация новых пользователей при помощи соц сетей.
  - 6.2. Авторизация существующих пользователей
  - 6.3. Личные профили пользователей с возможностью настройки личных данных и настроек сайта.
  - 6.4. Просмотр списка комиксов, с которыми взаимодействовал пользователь, уведомлений и оставленных комментариев
- 7. Социальная активность:
  - 7.1. Просмотр новостей, топа пользователей, последних тем на форуме
  - 7.2. Просмотр профилей других пользователей: их списки тайтлов, комментарии, друзья и пр.

- 7.3. Возможность добавить пользователя в друзья, написать сообщение
8. Мобильная адаптация и прочие возможности пользовательского интерфейса:
  - 8.1. Адаптация сайта для мобильных устройств и планшетов.
  - 8.2. Изменение темы оформления
9. Поддержка:
  - 9.1. Служба поддержки и обратной связи для пользователей

### Оценка трудоемкости работ (наивный метод)

№	Функционал	Оценка , мин. /чел. час	Оценк, вероятн . /чел. час	Оценка, макс. /чел. час
1	Раздел каталога	30	40	50
2	Раздел просмотра информации о тайтле	15	20	30
3	Раздел комментариев	20	30	40
4	Раздел оценок	10	20	30
5	Раздел просмотра обсуждений	15	20	30
6	Раздел редактирования манги (описание/главы)	10	20	30
7	Раздел списков (читаю/ в планах/ избранное и т.д.)	10	15	20
8	Режим чтения	40	50	60
9	Раздел регистрации авторизации	10	15	20
10	Раздел личного кабинета	20	30	40
11	Раздел главной страницы	15	20	25
12	Раздел новостей	20	30	40
13	Темы оформления	10	15	20
14	Мобильная адаптация	30	40	50
15	Раздел «Контакты»	10	15	20
16	Раздел добавления (автора, тайтла, команды)	10	15	20

17	Создание даталогической модели БД	10	12	14
18	Настройка git репозитория	6	8	10
19	Проектирование контрактов API	15	20	25
20	Базовая настройка окружения и подключение зависимостей (docker, frontend, backend)	8	10	12
21	Выбор стека технологий	12	15	18
22	Модульное тестирование	70	80	90
23	Интеграционное тестирование	110	120	130
24	Приемочное тестирование	50	60	70

### PERT-метод

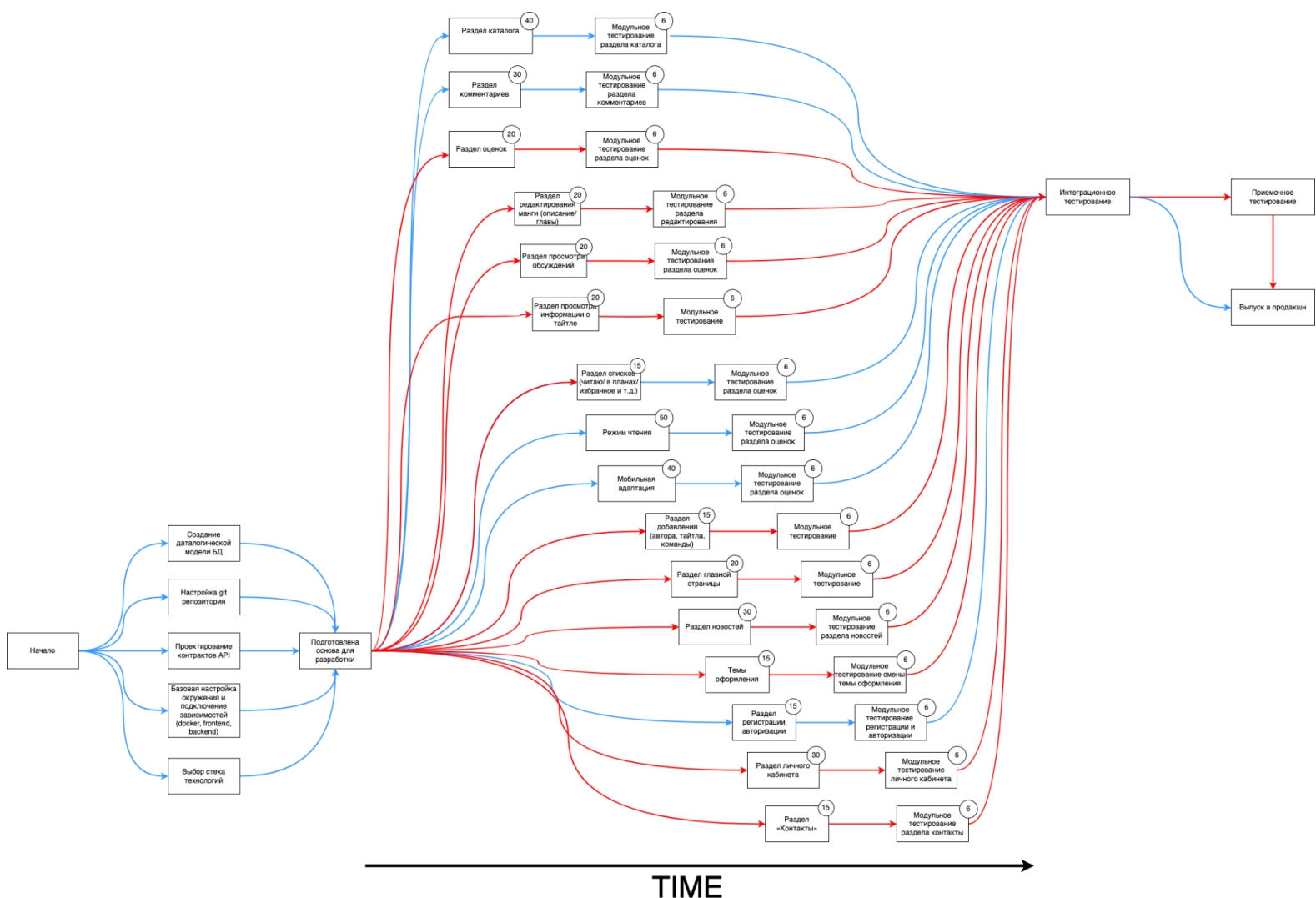
$$E_i = \frac{(P_i + 4M_i + O_i)}{6}$$

$$CKO_i = \frac{(P_i - O_i)}{6}$$

№	Функционал	Оценка, мин. /чел. час	Оценка, вероятн. /чел. час	Оценка, макс. /чел. час	$E_i$	$CKO_i$
1	Раздел каталога	30	40	50	40,0	3,3
2	Раздел просмотра информации о тайтле	15	20	30	20,8	2,5
3	Раздел комментариев	20	30	40	30,0	3,3
4	Раздел оценок	10	20	30	20,0	3,3
5	Раздел просмотра обсуждений	15	20	30	20,8	2,5
6	Раздел редактирования манги (описание/главы)	10	20	30	20,0	3,3

7	Раздел списков (читаю/ в планах/ избранное и т.д.)	10	15	20	15,0	1,7
8	Режим чтения	40	50	60	50,0	3,3
9	Раздел регистрации авторизации	10	15	20	15,0	1,7
10	Раздел личного кабинета	20	30	40	30,0	3,3
11	Раздел главной страницы	15	20	25	20,0	1,7
12	Раздел новостей	20	30	40	30,0	3,3
13	Темы оформления	10	15	20	15,0	1,7
14	Мобильная адаптация	30	40	50	40,0	3,3
15	Раздел «Контакты»	10	15	20	15,0	1,7
16	Раздел добавления (автора, тайтла, команды)	10	15	20	15,0	1,7
17	Создание даталогической модели БД	10	12	14	12,0	0,7
18	Настройка git репозитория	6	8	10	8,0	0,7
19	Проектирование контрактов API	15	20	25	20,0	1,7
20	Базовая настройка окружения и подключение зависимостей (docker, frontend, backend)	8	10	12	10,0	0,7

21	Выбор стека технологий	12	15	18	15,0	1,0
22	Модульное тестирование	70	80	90	80,0	3,3
23	Интеграционное тестирование	110	120	130	120,0	3,3
24	Приемочное тестирование	50	60	70	60,0	3,3



Среднее время разработки:

- по критическому пути: 455 часов
- по обычному пути: 720 часов



По предположительным расчетам на бэкенд вышло примерно 275 часов

На фронтенд 180 часов

Так как бэкенд объемнее чем фронтенд, то нам понадобится больше бэкенд разработчиков, чтобы эффективно распараллелить процесс разработки.

Общее количество человеко-часов не поменяется, зато абсолютное время уменьшится в (количество добавленных разработчиков) раз.

Поэтому мы приняли решение добавить 3 бэкенд и 2 фронтенд разработчиков на проект.

Если считать что в день работаем 8 часов, то команда справиться с проектом за

Бэкенд:  $275 / (8 * 3) \sim 12$  рабочих дней

Фронтенд:  $180 / (8 * 2) \sim 12$  рабочих дней

Учитывая, что фронтенд и бэкенд могут разрабатываться параллельно, то на весь проект по критическому пути уйдет 12 рабочих дней

А если бы на проекте работали бы по одному фронтенд и бэкенд разработчику, то потребовалось бы:

бэкенд  $275 / 8 \sim 35$  раб. дней

фронтенд  $180 / 8 \sim 23$  раб. дня

Таким образом, мы сократили абсолютное время разработки проекта на  $35 - 12 = 23$  рабочих дня.

Оценка размера проекта методом функциональных точек (FP).

Раздел каталога:

- Внешние входы (External Inputs):
  - Добавление новой манги в каталог.
  - Обновление информации о манге в каталоге.
- Внешние выходы (External Outputs):
  - Просмотр списка манги в каталоге.
- Внутренние логические файлы (Internal Logical Files):
  - Операции чтения и записи данных о манге.

#### Раздел комментариев:

- Внешние вводы:
  - Добавление комментария к манге.
- Внешние выводы:
  - Просмотр комментариев к манге.
- Внутренние логические файлы:
  - Операции чтения и записи комментариев.

#### Раздел списков:

- Внешние вводы:
  - Добавление манги в список "Читаю," "В планах," "Избранное," и т.д.
- Внешние выводы:
  - Просмотр списков манги.
- Внутренние логические файлы:
  - Операции чтения и записи данных о списках.

#### Режим чтения:

- Внешние вводы:
  - Навигация между страницами манги.
- Внутренние логические файлы:
  - Операции чтения и отображения текущей страницы манги.

#### Раздел регистрации/авторизации:

- Внешние вводы:
  - Регистрация нового пользователя.
  - Аутентификация пользователя.
- Внутренние логические файлы:
  - Операции чтения и записи данных о пользователях.

Оценочные коэффициенты могут быть числами от 0 до 5, где 0 означает низкую сложность, а 5 - высокую сложность.

#### Раздел каталога:

- Внешние вводы (External Inputs): 4 (средняя сложность)
- Внешние выводы (External Outputs): 3 (средняя сложность)
- Внутренние логические файлы (Internal Logical Files): 4 (средняя сложность)

#### Раздел комментариев:

- Внешние вводы: 3 (средняя сложность)
- Внешние выводы: 5 (высокая сложность)
- Внутренние логические файлы: 3 (средняя сложность)

#### Раздел списков:

- Внешние вводы: 4 (средняя сложность)
- Внешние выводы: 3 (средняя сложность)
- Внутренние логические файлы: 3 (средняя сложность)

#### Режим чтения:

- Внешние вводы: 3 (средняя сложность)
- Внутренние логические файлы: 2 (низкая сложность)

#### Раздел регистрации/авторизации:

- Внешние вводы: 4 (средняя сложность)
- Внутренние логические файлы: 3 (средняя сложность)

Чтобы определить суммарное количество не выровненных функциональных точек (UFP) необходимо сложить произведение количества операций в каждом разделе, на коэффициент сложности:

#### Раздел каталога:

- Внешние вводы (External Inputs):
  - Добавление новой манги в каталог:  $4 * 1 = 4$  UFP
  - Обновление информации о манге в каталоге:  $4 * 1 = 4$  UFP
- Внешние выводы (External Outputs):
  - Просмотр списка манги в каталоге:  $3 * 1 = 3$  UFP
- Внутренние логические файлы (Internal Logical Files):
  - Операции чтения и записи данных о манге:  $4 * 2 = 8$  UFP (2 операции)

#### Раздел комментариев:

- Внешние вводы:
  - Добавление комментария к манге:  $3 * 1 = 3$  UFP
- Внешние выводы:
  - Просмотр комментариев к манге:  $5 * 1 = 5$  UFP
- Внутренние логические файлы:
  - Операции чтения и записи комментариев:  $3 * 2 = 6$  UFP (2 операции)

#### Раздел списков:

- Внешние вводы:
  - Добавление манги в список "Читаю," "В планах," "Избранное," и т.д.:  $4 * 1 = 4$  UFP
- Внешние выводы:
  - Просмотр списков манги:  $3 * 1 = 3$  UFP
- Внутренние логические файлы:

- Операции чтения и записи данных о списках:  $3 * 2 = 6$  UFP (2 операции)

Режим чтения:

- Внешние вводы:
  - Навигация между страницами манги:  $3 * 1 = 3$  UFP
- Внутренние логические файлы:
  - Операции чтения и отображения текущей страницы манги:  $2 * 2 = 4$  UFP (2 операции)

Раздел регистрации/авторизации:

- Внешние вводы:
  - Регистрация нового пользователя:  $4 * 1 = 4$  UFP
  - Аутентификация пользователя:  $4 * 1 = 4$  UFP
- Внутренние логические файлы:
  - Операции чтения и записи данных о пользователях:  $3 * 2 = 6$  UFP (2 операции)

$$4 + 4 + 3 + 8 + 3 + 5 + 6 + 4 + 3 + 6 + 3 + 4 + 4 + 6 = 63 \text{ UFP}$$

После определения типов функциональных элементов и количества операций в каждом из них, можно применить оценочные коэффициенты, чтобы учесть сложность каждого элемента.

## Расчет трудоемкости методом COCOMO II.

Тип проекта

может быть "Organic," так как разработка сервиса для чтения манги, скорее всего, будет считаться небольшим или средним проектом.

Определение размера проекта в строках кода.

63 UFP

Для backend: java (53 avg) \* 63 = 3 339 строк

Для frontend: js (47 avg) \* 63 = 2961 строка

Расчет фактора коррекции EAF (Effort Adjustment Factor)

на основе характеристик проекта, таких как сложность, персонал, процессы и другие.

	Очень низкий	Низкий	Средний	Высокий	Очень высокий	Критический
Требуемая надежность ПО	0.75	0.88	1	1.15	1.4	-
Размер БД приложения	-	0.94	1	1.08	1.16	-
Сложность продукта	0.7	0.85	1	1.15	1.3	1.65
Требования к быстродействию	-	-	1	1.11	1.3	1.66
Ограничения памяти	-	-	1	1.06	1.21	1.56
Неустойчивость окружения	-	0.87	1	1.15	1.3	-
Время восстановления	-	0.87	1	1.07	1.15	-
Аналитические способности	1.46	1.19	1	0.86	0.71	-
Способности к разработке ПО	1.29	1.13	1	0.91	0.82	-
Опыт разработки	1.42	1.17	1	0.86	0.7	-
Опыт использования виртуальных машин	1.21	1.1	1	0.9	-	-
Опыт языка	1.14	1.07	1	0.95	-	-
Инструменты разработки	1.24	1.1	1	0.91	0.82	-
Методы разработки	1.24	1.1	1	0.91	0.83	-
График разработки	1.23	1.08	1	1.04	1.1	-

$$EAF = 0.226$$

Расчет трудоемкости проекта с использованием формулы  
COCOMO II

$$\text{Трудоемкость} = A * (\text{Размер проекта в строках кода})^B * EAF$$

- $A = 2.4$
- $B = 1.05$

$$\text{Трудоемкость} = 2.4 * (\text{Размер проекта в тыс. строках кода})^{1.05} *$$

$$EAF = 3,46 \text{ человеко - месяцев}$$

Расчет продолжительности проекта и необходимого количества  
человеко-месяцев

$$\text{Продолжительность} = C * (\text{Трудоемкость})^D \text{ Человеко-месяцы} =$$
$$\text{Трудоемкость} / \text{Продолжительность}$$

Для типа "Organic" COCOMO II, значения параметров могут быть:

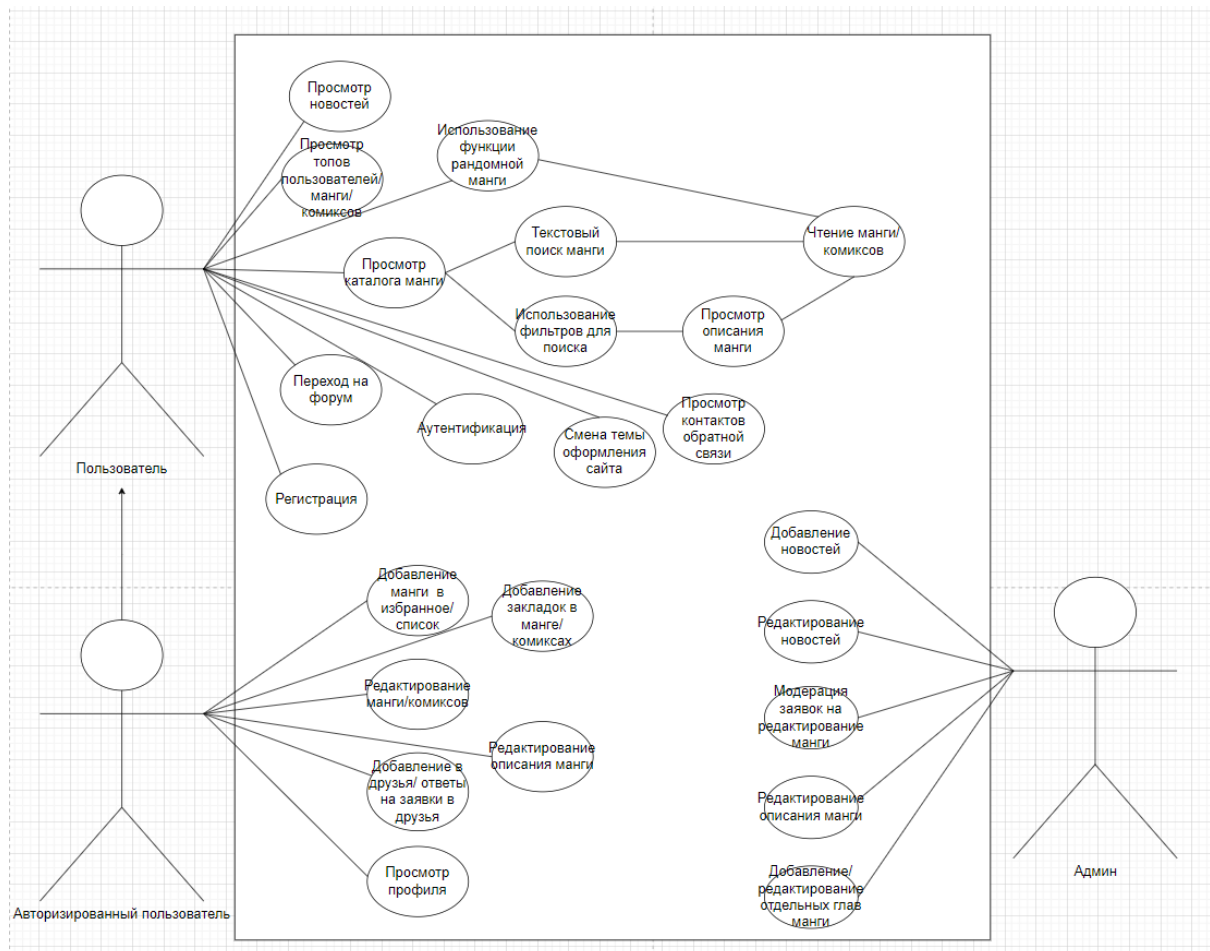
- $C = 2.5$
- $D = 0.38$

$$\text{Продолжительность} = 2.5 * (\text{Трудоемкость})^{0.38} = 4 \text{ месяца}$$

$$\text{Число разработчиков} = \text{Трудоемкость} / \text{Продолжительность} = 0.85$$

## Метод оценки вариантов использования

Use Case диаграмма



Невыровненный вес участника (UAW)

Классификация	Количество участников	Вес	Затраты
Простой	1	1	1
Средний	1	2	2
Сложный	1	3	3

UAW = 6

Невыровненный вес прецедентов (UUCW)

Классификация	Количество прецедентов	Вес	Затраты
Простой	16	5	80
Средний	4	10	40
Сложный	4	15	60

UUCW = 180

# Оценка фактора технической сложности (TCF)

Фактор	Вес (W)	Сложность (F)	Затраты
Децентрализованная архитектура	2	0	0
Время ответа / Производительность	1	2	2
Пользовательская эффективность	1	5	5
Внутренняя обработка данных	1	0	0
Переиспользование кода	1	0	0
Установка и настройка	0.5	1	0.5
Удобство использования	0.5	5	2.5
Переносимость на различные платформы	2	2	4
Обслуживание и поддержка	1	2	2
Параллельная обработка данных	1	1	1
Безопасность функций	1	0	0
Доступ для сторонних пользователей	1	3	3
Обучение конечных пользователей	1	0	0

TF = 20

TCF = 0.6 + (TF/100) = 0.8

## Оценка факторов окружающей среды (ECF)

Фактор	Описание	Вес	Влияние	Затраты
E1	Знакомство с	1.5	3	4.5



	процессом разработки, который был применен ранее.			
E2	Опыт применения данного процесса разработки.	0.5	4	2
E3	Опыт команды в объектно-ориентированной разработке.	1	1	1
E4	Компетенции ведущего аналитика.	0.5	1	0.5
E5	Мотивация участников команды.	1	2	2
E6	Устойчивость и ясность требований.	2	4	8
E7	Сотрудники, работающие неполный рабочий день.	-1	3	-3
E8	Использование сложных технологий	-1	1	-1

$$EF = 14$$

$$ECF = 1.4 + (-0.03 * EF) = 1.82$$

Подсчет фактора продуктивности на основе прошлого проекта. PF

В качестве примера взяли проект курсовой работы по ИСБД - Криптовалютный сервис  
Прецеденты проекта.

№	Описание
1	Регистрация и аутентификация
2	Пополнение баланса с карты
3	Обмен криптовалюты между своими счетами
4	Перевод криптовалюты на другой

	кошелек
5	Покупка НФТ на маркетплейсе
6	Оценка НФТ на маркетплейсе
7	R2P торговля
8	Просмотр текущих курсов криптовалют
9	Привязать банковскую карту

Невыровненный вес участника (UAW)

Классификация	Количество участников	Вес	Затраты
Простой	1	1	1
Средний	0	2	2
Сложный	1	3	3

UAW = 4

Невыровненный вес прецедентов (UUCW)

Классификация	Количество прецедентов	Вес	Затраты
Простой	2	5	10
Средний	2	10	20
Сложный	5	15	75

UUCW = 105

Оценка фактора технической сложности (TCF)

Фактор	Вес (W)	Сложность (F)	Затраты
Децентрализованная архитектура	2	2	4
Время ответа / Производительность	1	4	4
Пользовательская эффективность	1	3	3
Внутренняя обработка данных	1	0	0

Переиспользование кода	1	0	0
Установка и настройка	0.5	1	0.5
Удобство использования	0.5	5	2.5
Переносимость на различные платформы	2	2	4
Обслуживание и поддержка	1	3	3
Параллельная обработка данных	1	2	2
Безопасность функций	2	3	6
Доступ для сторонних пользователей	1	3	3
Обучение конечных пользователей	2	1	2

TF = 34

TCF = 0.6 + (TF/100) = 0.94

Оценка факторов окружающей среды (ECF)

Фактор	Описание	Вес	Влияние	Затраты
E1	Знакомство с процессом разработки, который был применен ранее.	1.5	3	4.5
E2	Опыт применения данного процесса разработки.	0.5	4	2
E3	Опыт команды в объектно-ориентированной разработке.	1	1	1
E4	Компетенции ведущего аналитика.	0.5	1	0.5
E5	Мотивация участников	2	2	4

	команды.			
E6	Устойчивость и ясность требований.	2	4	8
E7	Сотрудники, работающие неполный рабочий день.	0.5	2	1
E8	Использование сложных технологий	2	3	6

$$EF = 27$$

$$ECF = 1.4 + (-0.03 * EF) = 2.21$$

Подсчет UCP

$$UCP = (UCW + UAW) * TCF * ECF = 226$$

Подсчет трудоемкости проекта:

Предыдущая работа была выполнена за 200 часов на двух человек

$$PF = E/UCP = 0.88$$

UCP для нашего = 270

$$E = PF * UCP = 0.88 * 270 = 240$$

Добавим сюда юридическое согласование, требование безопасников,, интеграции и получим

$$240 + 150 + 50 = 440 \text{ часов}$$

## Сравнение использованных методов

Итоговая трудоемкость различными методами:

- Наивный метод: 720 человеко-часов
- PERT: 721 человека-часов  
по критическому пути: 455 часов
- МФТ+ СОСОМО-И: 640 часов
- ПСР: 440 человеко-часов

Видно, что все методы, кроме метода функциональных точек, дали примерно одинаковый результат. Отклонение метода функциональных точек связано с отсутствием оценки рисков и малым опытом команды.

Наивный метод позволил быстро оценить трудоемкость проекта, не проводя дополнительных вычислений. Однако видно, что его результат немного меньше получившихся с использованием других методов, и в целом, полагаться на него довольно трудно, так как его эффективность напрямую зависит от опыта участников.

Метод PEVT потребовал очень небольших трудовых затрат для проведения вычислений и позволил уточнить результаты наивного метода. С учетом низкой сложности вычислений использование PEVT кажется оправданным, однако он базируется на результатах наивного метода, потому по-прежнему напрямую зависит от опыта участников.

Метод функциональных точек дал результат, значительно отличающийся от других в меньшую сторону. Это связано с тем, что данный метод слабо приспособлен для небольших проектов, таких как наш. В результате, полученные значения слабо отражают реальные трудовые затраты, при том, что трудовые затраты на проведение оценки значительно превышают затраты на наивный и метод PEVT.

Метод CP дал результат, сопоставимый с другими методами. Его результаты кажутся наиболее близкими к реальности, в связи с более глубоким анализом проекта, чем в первых двух методах. При этом полученный результат выглядит гораздо более правдоподобным, нежели полученный по МФТ. К недостаткам метода можно отнести тот факт, что в полной мере он раскрывается при большом количестве проектов в качестве базы для оценки, поэтому на начальных этапах отклонение может быть большим, чем хотелось бы.

## Вывод

Во время выполнения лабораторной работы мы изучили различные методы оценки временных и ресурсных затрат на разработку проекта, а именно наивная оценка, PERT, метод функциональных точек, метод COCOMO II и User Case Points.