**Отчет**

Лабораторная работа № 1

«Экономика ПИ»

вариант: [https:/kaggle.com](https://kaggle.com)

**Выполнила:**

Дау Конг Туан Ань

Ван Минчжи

**Группа:**

Р33151

**Преподаватель:**

Гаврилов Антон Валерьевич

# **Вариант:**

[https:/kaggle.com](https://kaggle.com)

# **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.

**Каталог**

[**Вариант** 2](#_Toc179495274)

[**Задание** 2](#_Toc179495275)

[**Функциональные требования** 4](#_Toc179495276)

[**Наивый метод** 6](#_Toc179495277)

[**PERT метод** 9](#_Toc179495278)

[**Метод функциональной точки** 11](#_Toc179495279)

[**COCOMO II** 16](#_Toc179495280)

[**Use Case Points** 19](#_Toc179495281)

[**Вывод** 22](#_Toc179495282)

# **Функциональные требования**

**1. функции управления пользователями**

**1.1** Система должна обеспечивать функции регистрации и входа пользователей, в том числе:

1.1.1 Пользователи могут регистрироваться по электронной почте и паролю.

1.1.2 Система должна поддерживать быструю регистрацию и вход через социальные аккаунты (например, Google, Facebook, GitHub).

1.1.3 После того как пользователь вошел в систему, система должна предоставить страницу профиля пользователя, где он может изменить личную информацию.

1.1.4 Система должна поддерживать функцию двойной аутентификации для защиты учетной записи пользователя.

**1.2** Управление пользователями и разрешения:

1.2.1 Пользователи должны быть разделены на роли, такие как обычные пользователи, специалисты по исследованию данных, администраторы и т.д., в зависимости от их полномочий.

1.2.2 Система должна предоставлять функции управления пользователями для администраторов, включая включение/отключение учетных записей пользователей.

**2. Функции управления наборами данных**

**2.1** Система должна предоставлять функции загрузки и выгрузки наборов данных, включая:

2.1.1 Пользователи могут загружать пользовательские наборы данных.

2.1.2 Система должна предоставлять подробное описание загружаемых данных и функцию проверки формата.

2.1.3 Набор данных должен иметь функцию просмотра, чтобы пользователи могли просмотреть описание и примеры данных набора.

2.1.4 Набор данных должен обеспечивать функцию загрузки, пользователи могут загружать публичные или частные наборы данных в соответствии с полномочиями.

**2.2** Управление версиями наборов данных:

2.2.1 Система должна поддерживать загрузку нескольких версий наборов данных.

2.2.2 Изменения в каждой версии должны быть записаны, и пользователи могут просматривать исторические версии.

**3. Функции конкурса**

**3.1** Система должна обеспечивать функции создания и управления конкурсами, включая:

3.1.1 Пользователи (например, компании, организации) могут публиковать конкурсы по науке о данных.

3.1.2 Система должна поддерживать установку наград, правил и критериев оценки конкурса.

3.1.3 Система должна поддерживать указание наборов данных для конкурса.

**3.2** Участие в конкурсе:

3.2.1 Пользователи должны иметь возможность участвовать в открытых конкурсах.

3.2.2 Пользователи должны иметь возможность отправлять результаты конкурсов, а система будет автоматически оценивать их в соответствии с правилами конкурса.

3.2.3 Система должна отображать таблицу лидеров в реальном времени, показывающую результат и рейтинг пользователя.

**4. Управление сценариями и кодами**

**4.1** Система должна предоставлять функции обмена и выполнения сценариев кода, в том числе:

4.1.1 Пользователи могут загружать, обмениваться и запускать скрипты, например Python.

4.1.2 Система должна поддерживать несколько сред программирования (например, Jupyter Notebook) и обеспечивать функции выполнения в режиме онлайн.

4.1.3 Пользователи могут просматривать код других пользователей, оставлять комментарии и вести обсуждения.

**5. Функции взаимодействия с сообществом**

**5.1** Система должна предоставлять дискуссионные форумы и функции комментирования, в том числе:

5.1.1 Пользователи могут обсуждать на страницах конкурсов и наборов данных.

5.1.2 Система должна поддерживать функции «нравится», «комментировать» и «отвечать», где пользователи могут ставить «нравится» или отмечать полезные ответы.

**5.2** Система должна предоставлять платформу для обмена блогами и статьями, в том числе:

5.2.1 Пользователи могут размещать статьи и блоги, связанные с наукой о данных.

5.2.2 Система должна поддерживать функции категоризации и тегирования статей, чтобы пользователи могли легко их находить.

**6. доска лидеров и система вознаграждений**

**6.1** Система должна предоставлять функции ранжирования пользователей, включая:

6.1.1 Пользователи могут накапливать баллы по результатам конкурсов, в которых они участвуют.

6.1.2 Система должна обеспечивать глобальный и региональный рейтинг пользователей.

**6.2** Система должна поддерживать систему вознаграждений:

6.2.1 Пользователи могут зарабатывать призы, трофеи и значки, участвуя в конкурсах.

6.2.2 На страницах профилей пользователей должны отображаться их награды и достижения.

**7. безопасность и конфиденциальность данных**

**7.1** Система должна обеспечивать комплексные меры безопасности данных и защиты конфиденциальности, включая:

7.1.1 Наборы данных, загружаемые пользователями, должны быть защищены шифрованием.

7.1.2 Система должна соответствовать международным законам о конфиденциальности, таким как GDPR, и предоставлять функции удаления и анонимизации данных.

**8. интерфейс API**

**8.1** Система должна предоставлять API-интерфейсы для вызова внешних приложений, в том числе:

8.1.1 Пользователи могут запрашивать и загружать публичные наборы данных через API.

8.1.2 Система должна предоставлять API, связанные с соревнованиями, которые позволяют пользователям подавать результаты или запрашивать рейтинги через API.

# **Наивый метод**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Этап** | **№** | **Название** | **Описание** | **Optimistic** | **Pessimistic** | **Optimal** |
| **(h)** | **(h)** | **(h)** |
| **Подготовка** | 1 | Прототип сайта | Определить веб-сайт для науки о данных, определить общую цель веб-сайта, пользователей сервиса, определить источники дохода, разработать логотип | 50 | 100 | 70 |
| 2 | Выбор технологий (фундамент) | Важно определиться с используемыми технологиями: Frontend - ReactJS подойдет нам лучше всего, потому что информационный сайт с красивыми анимациями можно очень быстро на нем написать. Backend - простой Java сервер для аутентификации пользователей и запись на турниры. Создание собственных турниров не предусмотрено | 8 | 30 | 24 |
| 3 | Аренда хостинга | В интернете довольно много вариантов для столь простого сайта, поэтому нужно найти лучший по отношению цена/качество | 8 | 36 | 24 |
|  | | | 66 | 166 | 118 |
| **Дизайн** | 1 | проектирование бизнес-логики | Создавайте бизнес-процессы, используя рисунки и рабочие процессы UML. | 20 | 60 | 40 |
| 2 | дизайн интерфейса | Завершите чертежи Figma для различных компонентов, связав их для формирования полного рабочего процесса. | 40 | 80 | 70 |
| 3 | Проектирование API - конечные точки | Опишите конкретную цель, путь к ним, ограничения, видимость, привилегии при вызове, соответствующее место использования при использовании во внешнем интерфейсе с помощью swagger. | 30 | 50 | 40 |
|  | | | 90 | 190 | 150 |
| **Выполнение** | 1 | Регистрация\Вход\Выход | Зарегистрируйтесь, используя свой адрес электронной почты, и сохраните данные в базе данных. Вызывается при входе в систему | 20 | 50 | 40 |
| 2 | Поиск по содержимому | Удобный поиск контента на веб-страницах. | 50 | 90 | 70 |
| 3 | Раздел «Кто на Kaggle?» | Просмотр контента Kaggle | 30 | 50 | 40 |
| 4 | Раздел «Набор данных» | Он сохраняет высококачественные общедоступные наборы данных, доступные каждому, с большим объемом и богатым содержанием. | 50 | 90 | 80 |
| 5 | Раздел «Блокнот» | Содержит много содержимого записной книжки, онлайн-записную книжку, вы сможете узнать опыт других людей. | 50 | 90 | 70 |
| 6 | Раздел «Модель» | Содержит множество предварительно отработанных и готовых к развертыванию моделей машинного обучения | 60 | 90 | 80 |
| 7 | Раздел «Конкурс» | Просмотрите доступные соревнования, включая подробную информацию о требованиях, призах, целях и т. д. | 60 | 90 | 70 |
| 8 | Раздел «Написание решения» | Изучите передовые методы машинного обучения и узнайте, что работает, а что нет у ведущих конкурентов Kaggle. | 60 | 90 | 70 |
| 9 | раздел «Курс» | Для обучения доступны бесплатные практические курсы. | 80 | 130 | 120 |
| 10 | раздел «Обсуждение» | Вы можете задавать вопросы и кто-то на них ответит | 50 | 90 | 60 |
| 11 | Раздел «Еще» | Маленькая кнопка, чтобы скрыть или развернуть параметры. | 10 | 30 | 20 |
| 12 | Раздел «Рейтинг пользователей» | О мастерах в Kaggle можно узнать здесь. | 20 | 40 | 30 |
| 13 | Раздел «Документация» | Введение в использование Kaggle | 10 | 30 | 20 |
| 14 | Раздел «Провести конкурс» | Введение в соревнования, проводимые на Kaggle, и некоторые сведения. | 40 | 60 | 50 |
| 15 | Раздел «перемещение» | Представляем, как продвигаться в Kaggle | 20 | 50 | 30 |
| 16 | Раздел "KaggleX" | Подробное введение в то, что такое KaggleX, как стать KaggleX и перспективы. | 20 | 40 | 30 |
| 17 | Раздел «Помощь» | Описывает, как решать распространенные проблемы, как и большинство веб-сайтов. | 10 | 30 | 20 |
| 18 | Раздел «Принципы сообщества» | Описывает, что запрещено в сообществе. | 10 | 30 | 20 |
| 19 | раздел ссылок | Вы можете перейти в сообщество ВК, Facebook и Ins. | 10 | 40 | 30 |
| 20 | Поддержка мобильных устройств |  | 70 | 150 | 130 |
| 21 | Условия | Перейти к соответствующим положениям и условиям | 20 | 50 | 40 |
|  | | | 750 | 1410 | 1120 |
| **Тестирование** | 1 | Тестирование модуля | Проверьте стабильную работу каждого компонента | 50 | 90 | 75 |
| 2 | Функциональное тестирование | Проверьте стабильную работу модулей вместе, убедившись в отсутствии конфликтов данных между компонентами одного пользователя. | 30 | 50 | 40 |
| 3 | Тестирование безопасности | Проверьте безопасность системы, обеспечив отсутствие несанкционированного доступа к компонентам системы. | 10 | 30 | 25 |
| 4 | Тестирование производительности | Тестирование производительности системы на разных уровнях пропускной способности, определение лучших, худших и наиболее оптимальных точек нагрузки. | 10 | 20 | 15 |
|  | | | 100 | 190 | 155 |
| **Релиз** | 1 | Тестирование (alpha + beta) | Проверка основного функционала и корректного отображения сайта и его таблиц в различных верстках | 48 | 96 | 72 |
| 2 | SSL certificate | Self-Signed или отсутствие не подойдет, настройка https соединения | 5 | 9 | 8 |
| 3 | Выбор окружения | На выбранный хост скинуть скомпилированный html страничку с ангуляром, добавить Java сервер | 12 | 24 | 18 |
| 4 | Настройка | Настройка хоста под наши нужды, проверка общения клиента и сервера | 5 | 15 | 10 |
|  |  |  | 70 | 144 | 108 |

Затраты = 1651 ч/ч

# **PERT метод**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Optimistic** | **Pessimistic** | **Optimal** |  |  |
| **(h)** | **(h)** | **(h)** |
| 1.1 | Прототип сайта | 50 | 100 | 70 | 71,67 | 8,33 |
| 1.2 | Выбор технологий (фундамент) | 8 | 30 | 24 | 22,33 | 3,67 |
| 1.3 | Аренда хостинга | 8 | 36 | 24 | 23,33 | 4,67 |
| 2.1 | проектирование бизнес-логики | 20 | 60 | 40 | 40,00 | 6,67 |
| 2.2 | дизайн интерфейса | 40 | 80 | 70 | 66,67 | 6,67 |
| 2.3 | Проектирование API - конечные точки | 30 | 50 | 40 | 40,00 | 3,33 |
| 3.1 | Регистрация\Вход\Выход | 20 | 50 | 40 | 38,33 | 5,00 |
| 3.2 | Поиск по содержимому | 50 | 90 | 70 | 70,00 | 6,67 |
| 3.3 | Раздел «Кто на Kaggle?» | 30 | 50 | 40 | 40,00 | 3,33 |
| 3.4 | Раздел «Набор данных» | 50 | 90 | 80 | 76,67 | 6,67 |
| 3.5 | Раздел «Блокнот» | 50 | 90 | 70 | 70,00 | 6,67 |
| 3.6 | Раздел «Модель» | 60 | 90 | 80 | 78,33 | 5,00 |
| 3.7 | Раздел «Конкурс» | 60 | 90 | 70 | 71,67 | 5,00 |
| 3.8 | Раздел «Написание решения» | 60 | 90 | 70 | 71,67 | 5,00 |
| 3.9 | раздел «Курс» | 80 | 130 | 120 | 115,00 | 8,33 |
| 3.10 | раздел «Обсуждение» | 50 | 90 | 60 | 63,33 | 6,67 |
| 3.11 | Раздел «Еще» | 10 | 30 | 20 | 20,00 | 3,33 |
| 3.12 | Раздел «Рейтинг пользователей» | 20 | 40 | 30 | 30,00 | 3,33 |
| 3.13 | Раздел «Документация» | 10 | 30 | 20 | 20,00 | 3,33 |
| 3.14 | Раздел «Провести конкурс» | 40 | 60 | 50 | 50,00 | 3,33 |
| 3.15 | Раздел «перемещение» | 20 | 50 | 30 | 31,67 | 5,00 |
| 3.16 | Раздел "KaggleX" | 20 | 40 | 30 | 30,00 | 3,33 |
| 3.17 | Раздел «Помощь» | 10 | 30 | 20 | 20,00 | 3,33 |
| 3.18 | Раздел «Принципы сообщества» | 10 | 30 | 20 | 20,00 | 3,33 |
| 3.19 | раздел ссылок | 10 | 40 | 30 | 28,33 | 5,00 |
| 3.20 | Поддержка мобильных устройств | 70 | 150 | 130 | 123,33 | 13,33 |
| 3.21 | Условия | 20 | 50 | 40 | 38,33 | 5,00 |
| 4.1 | Тестирование модуля | 50 | 90 | 75 | 73,33 | 6,67 |
| 4.2 | Функциональное тестирование | 30 | 50 | 40 | 40,00 | 3,33 |
| 4.3 | Тестирование безопасности | 10 | 30 | 25 | 23,33 | 3,33 |
| 4.4 | Тестирование производительности | 10 | 20 | 15 | 15,00 | 1,67 |
| 5.1 | Тестирование (alpha + beta) | 48 | 96 | 72 | 72,00 | 8,00 |
| 5.2 | SSL certificate | 5 | 9 | 8 | 7,67 | 0,67 |
| 5.3 | Выбор окружения | 12 | 24 | 18 | 18,00 | 2,00 |
| 5.4 | Настройка | 5 | 15 | 10 | 10,00 | 1,67 |
|  | | | | | | |

𝐸i - оценка средней трудоемкости

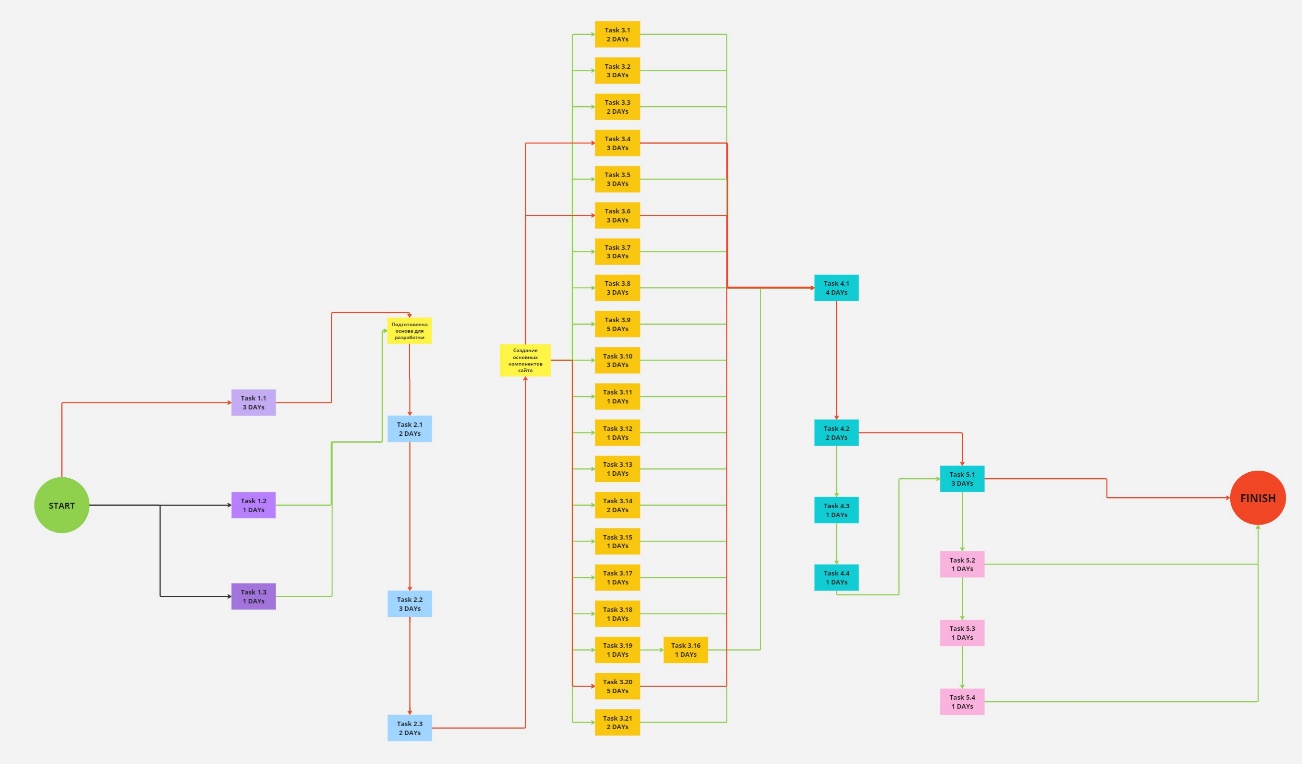
𝐸 - общая оценка статически независимых работ

СКОi - среднеквадратичное отклонение

СКО - среднеквадратичное отклонение для оценки суммарной трудоемкости

𝐸 - суммарная трудоемкость проекта (с вероятностью 95%)

# **Метод критического пути**



**Критический путь:** 487 ч./ч.

**Длинный путь:** 638 ч./ч.

**Выполнение проекта:** при ориентире на минимальное время разработки (критический путь) получаем, что для выполнения нам необходимо 487ч./ч.

**Команда:**

● 1x Аналитик

● 2x Fronend-разработчик

● 2x Backend-разработчик

● 2x Тестировщика

**Рабочий день считаем**: 8 часов (6 разработки + 1 обед + 1 тех. перерыв)

Таким образом, наша команда сможет выполнить проект за:

* Дизайн: 150 часов ( 25 раб. день)
* Выполнение: 1120 часов (187 раб. день)
* Тестирование: 155 часов ( 26раб. день)
* Релиз: 108 часов ( 18 раб. день)

**Время разработки**: 150+1120+155+108 = 1533 часов

**Общее время:** 25+187+26+18 = 256 рабочих дней

# [**Метод функциональной точки**](https://www2.seas.gwu.edu/~mlancast/cs254/FP%20Training.pdf)

**Порядок работы:**

1. Определите тип количества функциональных точек

2. Определите границу приложения

3. Определите и оцените типы транзакционных функций, чтобы определить их вклад в нескорректированное количество функциональных точек.

4. Определите и оцените типы функций данных, чтобы определить их вклад в нескорректированное количество функциональных точек.

5. Определите коэффициент корректировки значения (VAF)

6. Рассчитайте скорректированное количество функциональных точек.

1. **Определите тип количества функциональных точек**

Количество функциональных точек приложения: Оценивается объем уже существующего и установленного продукта

1. **Определите границу приложения**

Все функции. Рассчитываем все необходимые (реально используемые), а не дополнительные или только основные функции. Границы системы определены на UseCase диаграмме

1. **Определите и оцените типы транзакционных функций**

* **Record Element Type (RET):** это распознаваемая пользователем подгруппа элементов данных в ILF или EIF. Лучше всего взглянуть на логические группы данных, чтобы идентифицировать их.
* **Data Element Type (DET):** это уникальное, узнаваемое пользователем, нерекурсивное (неповторяющееся) поле. DET — это информация, которая является динамической, а не статической. Динамическое поле считывается из файла или создается из DET, содержащихся в FTR.
* **File Type Referenced (FTR)** : это тип файла, на который ссылается транзакция. FTR также должен быть внутренним логическим файлом или файлом внешнего интерфейса.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RET** | **DET** | **FTR** |
| Авторизоваться | 1 | 4 | 1 |
| Личный кабинет | 1 | 7 | 2 |
| форма создания набора данных | 1 | 5 | 2 |
| форма редактирования профиля | 2 | 9 | 3 |
| Форма создания нового проекта | 1 | 3 | 1 |

1. **Определите и оцените типы функций данных**

A table with text and check marks

Description automatically generated

* **External Inputs (EI) —** это элементарный процесс, в котором данные пересекают границу снаружи внутрь. Эти данные поступают извне приложения. Данные могут поступать с экрана ввода данных или из другого приложения. Данные могут использоваться для поддержания одного или нескольких внутренних логических файлов. Данные могут быть как управляющей информацией, так и деловой информацией.

A table with numbers and letters

Description automatically generated

* **External Outputs (EO)** — элементарный процесс, в котором полученные данные проходят через границу изнутри наружу. Кроме того, EO может обновлять ILF. Данные создают отчеты или выходные файлы, отправляемые в другие приложения. Эти отчеты и файлы создаются из информации, содержащейся в одном или нескольких внутренних логических файлах и файлах внешнего интерфейса.

A table with numbers and letters

Description automatically generated

* **External Inquiry (EQ)** — элементарный процесс с компонентами ввода и вывода, который приводит к извлечению данных из одного или нескольких внутренних логических файлов и файлов внешнего интерфейса. Процесс ввода не обновляет и не поддерживает никаких FTR (внутренних логических файлов или файлов внешнего интерфейса), а выходная сторона не содержит производных данных.

A table with text on it

Description automatically generated

* Internal Logical Files (ILF) — идентифицируемая пользователем группа логически связанных данных, которая находится полностью в границах приложения и поддерживается через внешние входы. Внутренний логический файл имеет неотъемлемое значение, он поддерживается внутри, имеет некоторую логическую структуру и хранится в файле.

A table with numbers and text

Description automatically generated

* External Interface Files (EIF) — идентифицируемая пользователем группа логически связанных данных, которая используется только для справочных целей. Данные находятся полностью за пределами границ приложения и поддерживаются внешними входами других приложений. Файл внешнего интерфейса — это внутренний логический файл для другого приложения

A table with numbers and letters

Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **EI** | **EO** | **EQ** | **EIF** | **ILF** |
| Авторизоваться | Low(3) | Low(4) | Low(3) | Low(5) | Low(7) |
| Личный кабинет | Average(4) | High(7) | Average(4) | Low(5) | Low(7) |
| форма создания набора данных | Average(4) | Average(5) | Low(3) | Low(5) | Low(7) |
| форма редактирования профиля | High(6) | High(7) | Average(4) | Low(5) | Low(7) |
| Форма создания нового проекта | Low(3) | Low(4) | Low(3) | Low(5) | Low(7) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Low** | **Average** | **High** | **Total** |
| EI | 6 | 8 | 6 | 20 |
| EO | 8 | 5 | 14 | 27 |
| EQ | 9 | 8 | 0 | 17 |
| EIF | 25 | 0 | 0 | 25 |
| ILF | 35 | 0 | 0 | 35 |

UFP =

1. **Определите коэффициент корректировки значения (VAF)**

Фактор корректировки стоимости (VAF) основан на 14 общих системных характеристиках (GSC), которые оценивают общую функциональность рассматриваемого приложения. Каждая характеристика имеет связанные описания для определения степени влияния.

Степень влияния варьируется по шкале от нуля до пяти, от отсутствия влияния до сильного влияния. Каждой характеристике присваивается рейтинг на основе подробных описаний, предоставленных Руководством IFPUG 4.1. Они оцениваются следующим образом:

0 Отсутствует или нет влияния

1 Случайное влияние

2 Умеренное влияние

3 Среднее влияние

4 Значительное влияние

5 Сильное влияние на протяжении всего

A text on a white background

Description automatically generated

A table of information

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **№** | **Параметр** | **Вес (DI)** |
| 1 | Обмен данными | 3 |
| 2 | Распределенная обработка данных | 0 |
| 3 | Производительность | 1 |
| 4 | Ограничения по аппаратным ресурсам | 3 |
| 5 | Транзакционная нагрузка | 0 |
| 6 | Интенсивность взаимодействия с пользователем | 4 |
| 7 | Эргономика | 3 |
| 8 | Интенсивность изменения данных | 1 |
| 9 | Сложность обработки | 0 |
| 10 | Повторное использование | 2 |
| 11 | Удобство инсталляции | 5 |
| 12 | Удобство администрирования | 1 |
| 13 | Портируемость | 1 |
| 14 | Гибкость | 0 |
|  | | |

1. **Рассчитайте скорректированное количество функциональных точек.**

AFP=UFP×VAF **=** 0.89 \* 124 = 110.36

Индекс AFP показывает, что сложность проекта находится на среднем уровне, поэтому можно оценить сроки и затраты на реализацию, а также правильно распределить ресурсы.

# [**COCOMO II**](https://www.cs.montana.edu/courses/spring2004/352/public/cocomo/modelman.pdf)

Оценка размера программного продукта в KSLOC:

**Стек технологий:**

* Reactjs (JS)
* Backend (Java)

Разделим функциональность между слоями: Подсчитаем размер 0.5 frontend и 0,5 backend по KSLOC:

**Факторы масштаба проекта:**

* PREC - прецедентность, наличие опыта аналогичных разработок
* FLEX - гибкость процесса разработки
* RESL - архитектура и разрешение рисков
* TEAM - сработанность команды
* PMAT - зрелость процессов

A table with text on it

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Название фактора** | **Уровень фактора** | **Значение уровня** |
| PREC | Nominal | 2.43 |
| FLEX | Very High | 1.21 |
| RESL | Very Low | 4.22 |
| TEAM | High | 1.98 |
| PMAT | Nominal | 2.73 |

Драйверы затрат:

Для предварительной оценки проекта необходимо оценить уровень семи множителей трудоемкости M:

* PERS - квалификация персонала
* RCPX - сложность и надежность продукта
* RUSE - разработка для повторного использования
* PDIF - сложность платформы разработки
* PREX - опыт персонала
* FCIL - оборудование
* CSED - требуемое выполнение графика работ

|  |  |  |
| --- | --- | --- |
| **Название** | **Уровень** | **Значение** |
| PERS | Very High | 0.63 |
| RCPX | Very High | 1.91 |
| RUSE | High | 1.07 |
| PDIF | Nominal | 1.00 |
| PREX | High | 0.87 |
| FCIL | Nominal | 1.0 |
| CSED | Low | 1.14 |

*В этой части правильный способ сделать это — рассчитать с помощью этой таблицы:*

*A list of cost driver

Description automatically generated*

*Но мы предполагаем, что данные по проекту не собираются, поэтому мы можем только оценить значение для каждой точки.*

Оцените усилия с помощью:

A math equations and formulas

Description automatically generated

A math equations on a white background

Description automatically generated

Где:

**A table with text and symbols

Description automatically generated with medium confidence**

DM =CM =IM = 0, AAF = 0, AA = 0, BRAK = 0, Итак, у нас есть:

= 25.41 ч. /мес

# **Use Case Points**

A diagram of a person with many lines and dots

Description automatically generated with medium confidenceA diagram of a person with text

Description automatically generated

|  |
| --- |
| UCP = TCP \* ECF \* UUCP \* PF |

1. Technical Complexity Factor (TCF).
2. Environment Complexity Factor (ECF).
3. Unadjusted Use Case Points (UUCP).
4. Productivity Factor (PF).

Необходимые шаги для создания оценки на основе метода UCP:

* Определение и вычисление технических факторов.
* Определение и вычисление экологических факторов.
* Вычисление нескорректированных точек вариантов использования.
* Определение фактора производительности.
* Вычисление произведения переменных.

**Technical Complexity Factors:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Технический фактор** | **Описание** | **Масса** | **Воспринимаемая сложность** | **Рассчитанный фактор** |
| T1 | Распределенная система | 2 | 3 | 6 |
| T2 | Производительность | 1 | 4 | 4 |
| T3 | Эффективность для конечного пользователя | 1 | 3 | 3 |
| T4 | Сложная внутренняя обработка | 1 | 5 | 5 |
| T5 | Возможность повторного использования | 1 | 3 | 3 |
| T6 | Легко установить | 0.5 | 5 | 2.5 |
| T7 | Легко использовать | 0.5 | 5 | 2.5 |
| T8 | Портативный | 2 | 4 | 8 |
| T9 | Легко изменить | 1 | 2 | 2 |
| T10 | Параллельный | 1 | 3 | 3 |
| T11 | Специальные функции безопасности | 1 | 2 | 2 |
| T12 | Предоставляет прямой доступ третьим лицам | 1 | 5 | 5 |
| T13 | Требуются специальные условия для обучения пользователей. | 1 | 1 | 1 |
|  | | | **Общий фактор** | **47** |

TCF = 0.6 + (0.1\* **Общий фактор) = 1.07**

**Environmental Complexity Factors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Фактор окружающей среды** | **Описание** | **Масса** | **Воспринимаемая сложность** | **Рассчитанный фактор** |
| E1 | Знакомство с UML | 1.5 | 4 | 6 |
| E2 | Опыт применения | 0.5 | 2 | 1 |
| E3 | Объектно-ориентированный опыт | 1 | 5 | 5 |
| E4 | Возможности ведущего аналитика | 0.5 | 2 | 1 |
| E5 | Мотивация | 1 | 1 | 1 |
| E6 | Требования к стабильности | 2 | 5 | 10 |
| E7 | Неполный рабочий день | -1 | 0 | 0 |
| E8 | Сложный язык программирования | 2 | 1 | 2 |
|  | | | **Общий фактор** | **26** |

ECF = 1.4 + (-0.03 \* **Общий фактор) = 0.62**

**Unadjusted Use Case Points (UUCP)**

***Unadjusted Use Case Weight* (UUCW):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тип варианта использования** | **Описание** | **Масса** | **Количество вариантов использования** | **Результат** |
| Простой | Простой пользовательский интерфейс, затрагивающий только один объект базы данных; его успешный сценарий состоит из 3 шагов или менее; его реализация включает менее 5 классов. | 5 | 20 | 100 |
| Средний | Больше дизайна интерфейса, затрагивающего 2 или более сущностей базы данных; от 4 до 7 шагов; его реализация включает от 5 до 10 классов. | 10 | 3 | 30 |
| Сложный | Включает в себя сложный пользовательский интерфейс или обработку и затрагивает 3 или более сущностей базы данных; более семи шагов; его реализация включает более 10 классов. | 15 | 4 | 60 |
|  |  |  | **Общий UUCW** | **190** |

***Unadjusted Actor Weight* (UAW):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тип актера** | **Описание** | **Масса** | **Количество актеров** | **Результат** |
| Простой | Актер представляет собой еще одну систему с определенным API. | 1 | 2 | 2 |
| Средний | Актер представляет собой еще 2 системы, взаимодействующие через протокол, например TCP/IP. | 2 | 2 | 4 |
| Сложный | Актер — это человек, взаимодействующий через интерфейс. | 3 | 1 | 3 |
|  |  |  | **Общий UAW** | **9** |

UUCP = UUCW + UAW = 190 + 9 = 199

**Productivity Factor:**

PF = 20

Анализ результатов

|  |  |
| --- | --- |
| **Метод** | **Затраты (h-h)** |
| Наивный | 1651 |
| PERT | 1852 |
| Функциональных точек | 1650 |
| COCOMO II | 4065 |
| UCP | 2640 |

# **Вывод**

С помощью 5 методов расчета трудоемкости создания сайта kaggle.com, хотя 5 метода дают разные результаты, все они показывают, что проект средний, поэтому вопросы задавайте часто, время строительства большое, поэтому строительный блок. необходимо иметь оптимальную стратегию строительства, а также подготовить ресурсы для завершения строительства.