**Отчет**

Лабораторная работа № 1

«Экономика ПИ»

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

**Выполнила:**

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**

**Выполнение**

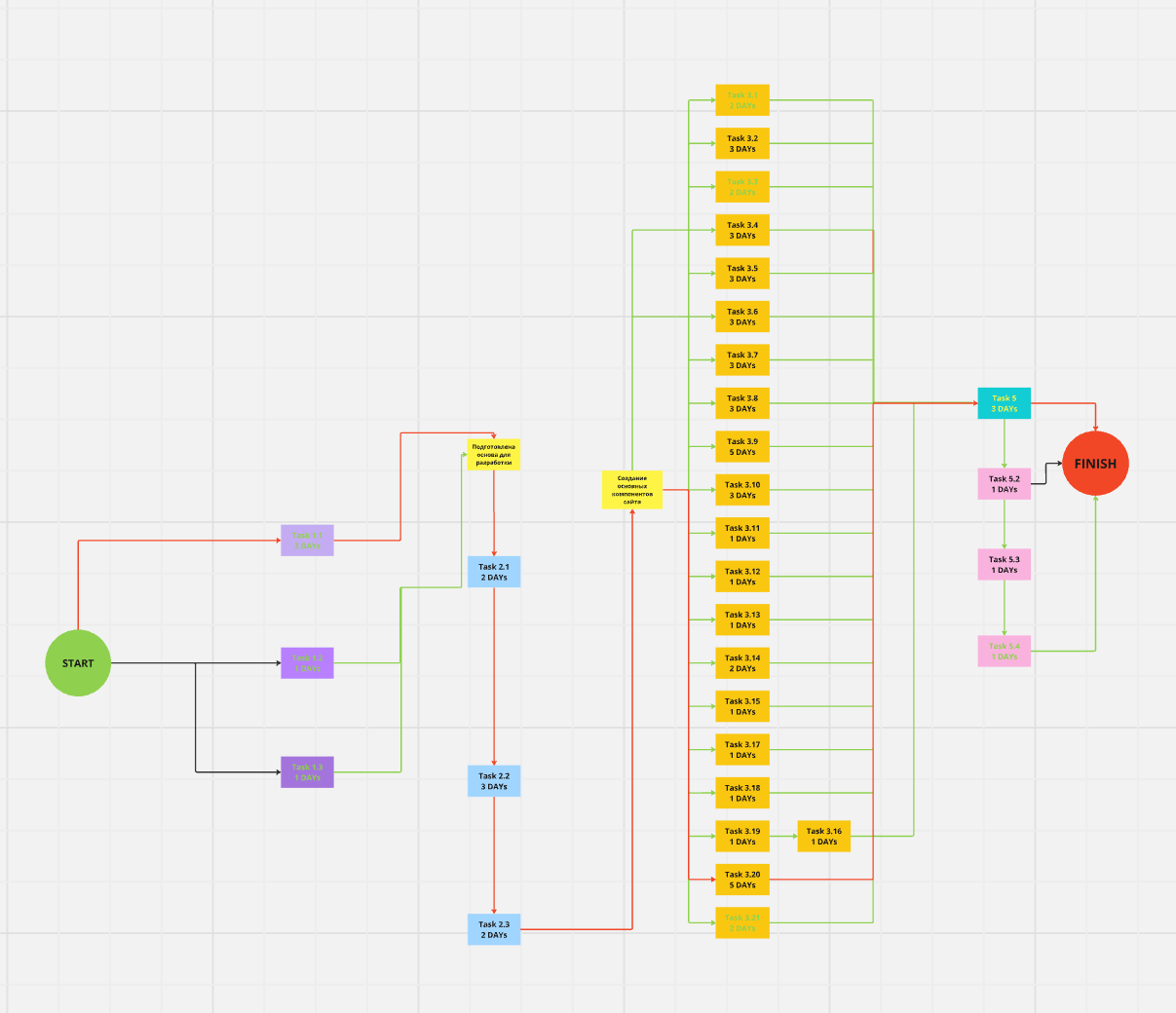
План действий при задержке

|  |  |
| --- | --- |
| 398h | задачи, которые согласно плану разработки выполнены в первую  половину периода разработки |
| 36h | задачи, на которые решили сократить время выполнения |
| 175h | задачи, от которых чего можно отказаться. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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

# **Метод критического пути**



**Оценка возможностей**

Основная часть функциональности была реализована.

对于测试部分进行了调整，减少了测试的1-4小节：单元测试、功能测试和安全测试的时间投入。当前版本的数据交互和事务量较低，只需保证核心功能能够顺利运行即可。我们保留了Alpha和Beta测试，但将时间缩减了一半，用于验证关键模块的基本功能是否符合预期。此外，由于现代浏览器对安全性的严格要求，SSL证书仍然保留。  
通过这次优化，总计节省了约211小时，使得项目在当前时间范围内完成成为可能，同时不会显著影响用户体验。

Тестовая часть была скорректирована для сокращения затрат времени на разделы 1-4 теста: модульное тестирование, функциональное тестирование и тестирование безопасности. Текущая версия имеет низкий уровень взаимодействия с данными и объем транзакций, и ей необходимо только обеспечить бесперебойную работу основных функций. Мы сохранили альфа- и бета-тестирование, но вдвое сократили время, чтобы проверить, соответствует ли базовая функциональность ключевых модулей ожидаемому. Кроме того, из-за строгих требований безопасности современных браузеров сертификаты SSL по-прежнему сохраняются.

Благодаря этой оптимизации в общей сложности было сэкономлено около 290 часов, что позволило завершить проект в текущие сроки без существенного влияния на пользовательский опыт.

**Вывод**

В этом лабе провела углубленный анализ и оптимизацию цикла разработки проекта и распределения задач. Основываясь на фактическом прогрессе и ограничениях ресурсов, мы провели приоритетную оценку каждой задачи и приняли следующие стратегии: классификация и оптимизация задач, гарантия основных функций, ключевые технологии и гарантия безопасности, что позволило выпустить проект в течение ограниченного времени.