|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_«Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_ «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 7**

|  |  |
| --- | --- |
| **Дисциплина** Экономика программной инженерии  **Тема** «Предварительная оценка параметров программного проекта»  **Студент** \_Ильясов И.М., Лаврова А.А., Игнатьев А.И.\_  **Группа** \_ИУ7-83/85Б\_  **Преподаватель** \_Барышникова М.Ю., Силантьева А.В.\_ |  |

Москва, 2021 г.

**Задание (вариант №2)**

Компания получила заказ на разработку клиентского мобильного приложения брокерской системы. Программа позволяет просматривать актуальную биржевую информацию, производить сделки и отслеживать их выполнение.

**Расчёт по методу функциональных точек**

Произведем расчет количества функциональных точек.

* **FTR** – количество связанных с каждым функциональным типом файлов типа ссылок.
* **DET** – количество связанных с каждым функциональным типом элементарных данных. (количество типов элементов данных)
* **RET** – количество типов элементов записей.
* **EI** (Внешний ввод) — элементарный процесс, перемещающий данные из внешней среды в приложение.
* **EO** (Внешний вывод) — элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду.
* **EQ** (Внешний запрос) — элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных или обновлением внутренних логических файлов (базы данных).
* **ILF** (Внутренний логический файл) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, которые поддерживаются внутри продукта и обслуживаются через внешние вводы.
* **EIF** (Внешний интерфейсный файл) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта

В нашем приложении используются 3 внутренних файла: таблица с логинами и паролями; таблица с типом заявки, именем бумаги, ценой и количеством; таблица с названием бумаги. Также существует одна внешняя таблица с информацией о бирже с названием бумаги, ценой и изменением.

**Вычислим EI (внешний ввод)**:

* Добавить бумагу:
  + FTR = 1 (ссылается на один внутренний логический файл)
  + DET = 2 (элементы данных: кнопка, название бумаги)
* Удалить заявку:
  + FTR = 1 (ссылается на один внутренний логический файл)
  + DET = 5 (элементы данных: тип, имя, цена, количество, кнопка)
* Изменить заявку:
  + FTR = 1 (ссылается на один внутренний логический файл)
  + DET = 5 (элементы данных: тип, имя, цена, количество, кнопка)
* Создать заявку:
  + FTR = 1 (ссылается на один внутренний логический файл)
  + DET = 5 (элементы данных: тип, имя, цена, количество, кнопка)

Уровень сложности – низкий

**Вычислим EO (внешний вывод)**:

* Вывод списка заявок:
  + FTR = 1 (ссылается на один внутренний логический файл)
  + DET = 4 (элементы данных: тип, имя, цена, количество)
* Вывод биржевых сводок:
  + FTR = 2 (ссылается на один внутренний логический файл и один внешний интерфейсный файл)
  + DET = 3 (элементы данных: имя, цена, изменения)

Уровень сложности – низкий.

**Вычислим EQ (внешний запрос)**:

* Внешний запрос на авторизацию:
  + FTR = 1 (ссылается на один внутренний логический файл)
  + DET = 4 (элементы данных: логин, пароль, кнопка, флажок)

Уровень сложности – низкий.

**Вычислим ILF (внутренний логический файл)**:

* RET = 4 (различные типы элементов записи)
* DET = 4 (различные типы элементов данных)

Уровень сложности – низкий.

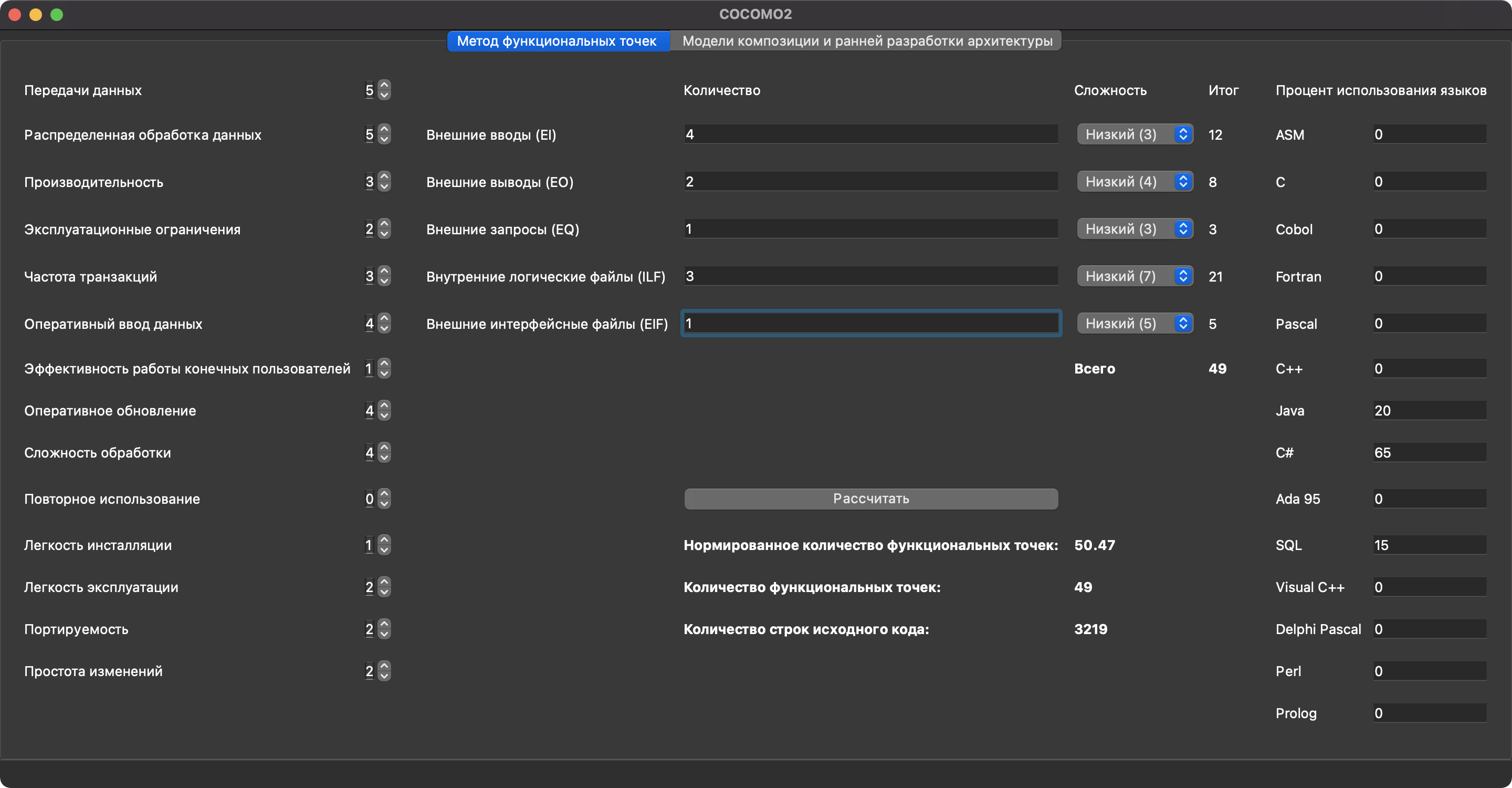
**Вычислим EIF (внешний интерфейсный файл)**:

* RET = 2 (различные типы элементов записи)
* DET = 3 (различные типы элементов данных)

Уровень сложности – низкий.

На рисунке 1 представлен результат:

* нормированное количество функциональных точек = 50.47;
* количество функциональных точек = 49;
* количество строк исходного кода = 3219.

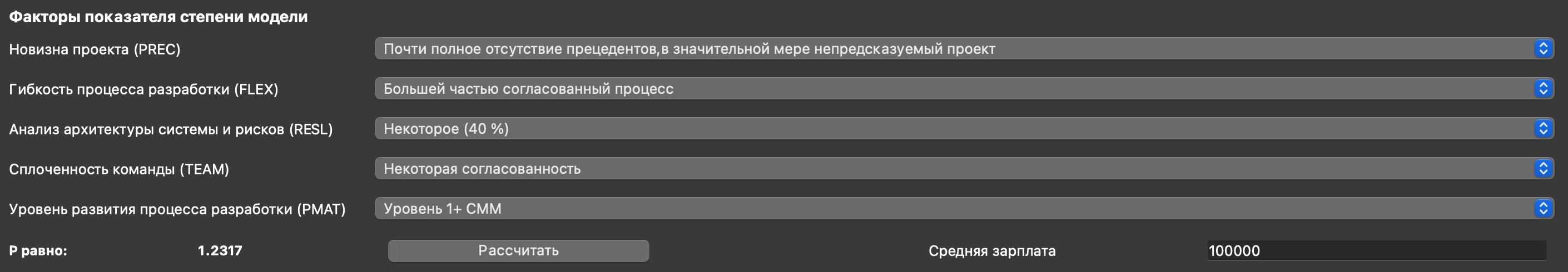
  
Рисунок 1 – Метод функциональных точек.

**Оценка по методике COCOMO II**

**Определим показатели проекта**:

* Новизна проекта (PREC) – полное отсутствие прецедентов, полностью непредсказуемый проект (т.к. была сформирована новая команда разработчиков, только отдельные члены имели некоторый опыт создания систем подобного типа).
* Гибкость процесса разработки (FLEX) – большей частью согласованный процесс (график жесткий, точной регламентации нет).
* Разрешение рисков в архитектуре системы (RESL) – некоторое (40%).
* Сплоченность команды (TEAM) – некоторая согласованность (команд новая, но были проведены определенные мероприятия по сплочению).
* Уровень развития процесса разработки (PMAT) – уровень 1+ (только начинают внедрять).

На рисунке 2 показан результат расчёта показателя степени: p = 1.2317.

  
Рисунок 2 – Факторы показателя степени модели

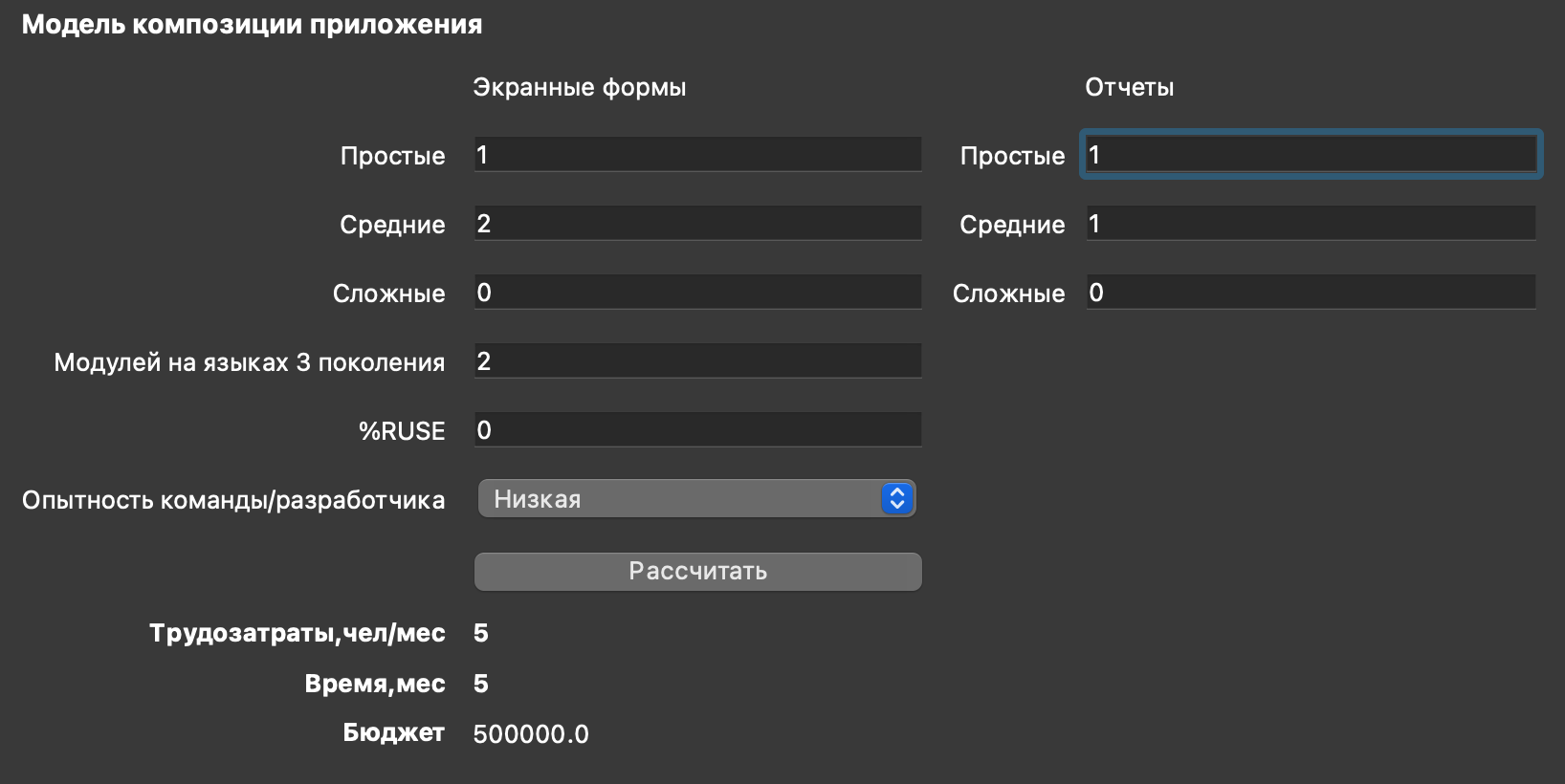
**Рассчитаем модель композиции приложения**:

* Страница авторизации – 1 экранная форма средней сложности.
* Страница биржевых сводок – 1 простая экранная форма и 1 отчет средней сложности.
* Страница заявок – 1 простой отчет.
* Страница новой заявки – 1 экранная форма средней сложности.

**Итого:**

* Простые экранные формы = 1
* Экранные формы средней сложности = 2
* Простые отчеты = 1
* Отчеты средней сложности = 1
* Также имеются 2 модуля, написанные на ЯП третьего поколения.
* Повторное использование = 0%
* Опытность команды – низкая

Результат работы программы представлен на рисунке 3.

  
Рисунок 3 – Модель композиции приложения

В итоге получилось, что трудозатраты равны 5 чел/мес, а время выполнения проекта равно 5 месяцам. При этом бюджет составил 500000 у.е. при средней зарплате 100000 у.е.

**Рассчитаем модель ранней разработки архитектуры**:

* PERS (возможности персонала) – номинальный
* RCPX (надежность и уровень сложности разрабатываемой системы) – очень высокий
* RUSE (повторное использование компонентов) – низкий
* PDIF (сложность платформы разработки) – высокий
* PREX (опыт персонал) – низкий
* FCIL (средства поддержки) – очень высокий
* SCED (график работ) – очень высокий
* KSLOC = 3.219 (из метода функциональных точек)

Результаты расчетов представлены на рисунке 4.

  
Рисунок 4 – Модель ранней разработки архитектуры

В итоге получилось, что трудозатраты равны 22 чел/мес, а время выполнения проекта равно 10 месяцам. При этом бюджет составил 2200000 у.е. при средней зарплате 100000 у.е.

**Вывод:**

В ходе выполнения данной работы был разработан инструмент для определения трудозатрат и времени разработки проекта методом COCOMO2. Также, был выполнен анализ выданного задания, а именно:

* рассчитаны функциональные точки;
* рассчитан показатель степени модели (p);
* были определены факторы, влияющие на показатель степени;
* произведен расчет трудозатрат и времени по модели ранней разработки архитектуры приложения и модели композиции приложения.

В итоге было выяснено, что модель композиции приложения дает намного более оптимистичный прогноз, по сравнению с моделью ранней архитектуры приложения. Связано это с тем, что в модели композиции приложения не учитывается информация о персонале, работающем над проектом в отличие от модели ранней архитектуры. Таким образом, можно сказать, что данная модель дает идеальный результат при условии наивысшей опытности команды и при идеальном протекании работы над проектом.