|  |  |
| --- | --- |
| 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»\_\_\_\_\_\_\_\_\_\_\_

**Отчет по лабораторной работе № 7**

|  |  |
| --- | --- |
| **По курсу: «**Экономика программной инженерии»  **Тема**  «Предварительная оценка параметров программного проекта»  **Студент**  Шингаров И. Д.  **Студент**  Блохин Д. М.  **Преподаватель**  Барышникова М.Ю.  **Преподаватель**  Силантьева А.В.  Москва, 2023 г. |  |

**Цель работы**

Продолжение знакомства с существующими методиками предварительной

оценки параметров программного проекта и практическая оценка затрат по

модели COCOMO II.

**Метод функциональных точек**

В этом методе рассматривается не размер, а функциональность или полезность продукта, в качестве количественной характеристики применяется понятие количества функциональных точек.

* *Количество функциональных точек* – количественная характеристика, обозначается как FP (function points).
* *Функциональная точка* — единица измерения функциональности ПО. Функциональность программы связана с обработкой информации по запросу пользователя и не зависит от применяемых технических решений.
* *Пользователи* — отправители и целевые получатели данных (люди, смежные интегрированные информационные системы).
* Оценка трудозатрат может быть выполнена на ранних стадиях работы над проектом, поскольку в основе лежит изучение требований.

Функциональность системы определяется:

1. Функциональными типами.
2. Элементарными процессами.

FTR – количество связанных с каждым функциональным типом файлов типа ссылок.

DET – количество связанных с каждым функциональным типом элементарных данных.

RET – количество типов элементов записей.

Типы элементарных процессов, используемых в методе функциональных точек:

* внешний ввод (EI) — элементарный процесс, перемещающий данные из внешней среды в приложение;
* внешний вывод (ЕО) — элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду;
* внешний запрос (EQ) — элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных;
* внутренний логический файл (ILF) — выделяемые пользователем логически связанные группы данных, которые поддерживаются внутри продукта и обслуживается через внешние вводы;
* внешний интерфейсный файл (EIF) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта.

**Постановка задачи**

Компания получила заказ на разработку клиентского мобильного приложения

брокерской системы. Программа позволяет просматривать актуальную

биржевую информацию, производить сделки и отслеживать их выполнение.

Приложение имеет 4 страницы: авторизация, биржевые сводки, заявки, новая

заявка.

**Характеристики команды, продукта и проекта**

Разработанное ПО состоит из трех компонентов. Первый компонент составляет по объему примерно 15% программного кода и будет написан на SQL, второй (около 60% кода) - на С#, а третий в объеме 25% кода - на Java.

*Характеристики продукта в условных единицах:*

1. Обмен данными - 5
2. Распределенная обработка -5
3. Производительность -3
4. Эксплуатационные ограничения по аппаратным ресурсам – 2
5. Транзакционная нагрузка – 3
6. Интенсивность взаимодействия с пользователем (оперативный ввод данных) – 4
7. Эргономические характеристики, влияющие на эффективность работы конечных пользователей – 1
8. Оперативное обновление – 4
9. Сложность обработки – 4
10. Повторное использование – 0
11. Легкость инсталляции – 1
12. Легкость эксплуатации/администрирования – 2
13. Портируемость – 2
14. Гибкость – 2

Для реализации проекта была сформирована новая команда разработчиков, у отдельных членов которой имеется некоторый опыт создания систем подобного типа. В целях сплочения команды были проведены определенные мероприятия,что обеспечило на старте проекта приемлемую коммуникацию внутри коллектива. Заказчик не настаивает на жесткой регламентации процесса, однако график реализации проекта довольно жесткий. Несмотря на то, что предметная область является для разработчиков относительно новой, анализу архитектурных рисков было уделено лишь некоторое внимание. Организация только начинает внедрять методы управления проектами и формальные методы оценки качества процесса разработки.

* Надежность и уровень сложности (RCPX) разрабатываемой системы оцениваются как очень высокие.
* Повторное использования компонентов не предусматривается (RUSE).
* Возможности персонала (PERS) – средние.
* Опыт работы в разработке систем подобного типа (PREX) низкий.
* Сложность платформы (PDIF) высокая.
* Разработка предусматривает очень интенсивное использование
* инструментальных средств поддержки (FCIL).
* Заказчик настаивает на жестком графике (SCED).

**Авторизация**

На данной странице осуществляется ввод логина и пароля пользователя для входа в систему. Страница содержит два поля ввода и одну командную кнопку, а также флажок-переключатель, который активируется при необходимости запоминания параметров авторизации.

**EIF:**

Таблица в базе данных (логин, пароль)

RET = 1 (логин, пароль – строка)

DET = 2 (логин, пароль)

**ILF:**

Объект локального хранилища (сохраненные данные)

RET = 1 (логин, пароль – строка)

DET = 2 (логин, пароль)

**EI:** Ввод данных, запись

FTR = 2 (2 файла); DET = 4 (логин, пароль, флаг, кнопка)

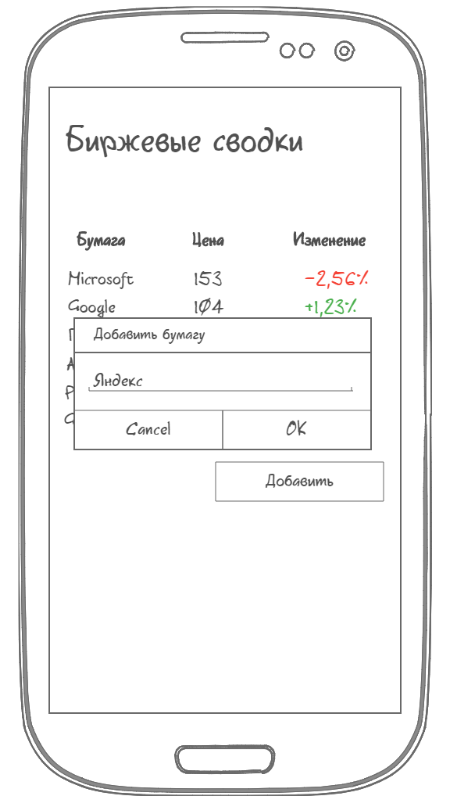
**EO:** Сообщение о некорректных данных

FTR = 1; DET = 4

**Биржевые сводки**

Биржевые сводки отражают текущую ситуацию на бирже. Страница содержит таблицу, кнопку «Добавить» и диалоговое окно с одним полем для ввода и двумя командными кнопками.

Таблица содержит три колонки: Ценная бумага (имя бумаги), Цена (цена за одну ценную бумагу), Изменение (изменение цены бумаги со времени последнего закрытия биржи). Кнопка «Добавить» вызывает диалоговое окно для добавления новой бумаги (окно состоит из поля ввода и кнопок ОК, Cancel).



**ILF:** Кэшированные данные таблицы

RET = 2 (строка, вещественное число с фиксированной запятой)

DET = 3 (бумага, цена изменение)

**EIF:** Таблица бумаг в БД биржи

RET = 2 (строка, вещественный)

DET = 2 (бумага, цена)

**EI:** Добавление новой бумаги

FTR = 1 Обновление списка интересующих бумаг

DET = 4 (“Добавить”, название, “Ok”, “Cancel”)

**EO:** Вывод цен и изменений

FTR = 2 (Изменение локальной таблицы и обращение к бирже)

DET = 3 (бумага, цена, изменение)

**EQ:** Внешний запрос к бирже

FTR = 1; DET = 4

**Заявки**

Заявки содержат таблицу, отображающую текущие (еще не выполненные) заявки на покупку или продажу ценных бумаг.

Таблица содержит четыре поля: Тип (покупка/продажа), Имя бумаги, Цена по которой готовы покупаться/продаваться бумаги, Количество бумаг для покупки/продажи.

При нажатии на любую строку таблицы появляется контекстное меню с возможностью удалить или изменить заявку.

**ILF:** Список открытых заявок

RET = 4 (символьный, строковый, вещественный и целый)

DET = 4 (тип, имя, кол-во, цена)

**EIF:**

***Таблица заявок на бирже***

RET = 4 (идентификаторы бумаги и покупателя (**строка** или целое число), вещественное число, символьный тип, целое число)

DET = 5 (покупатель(id), бумага(id), цена, тип, кол-во)

***Tаблица клиентов биржи***

RET = 2 (строка)

DET = 2 (идентификатор, логин)

***Таблица бумаг***

RET = 2 (строка, вещественный)

DET = 3 (идентификатор, бумага, цена)

**EI:**

***Изменить заявку***

FTR = 4 (Таблица бумаг, таблица пользователей, таблица заявок, локальное хранилище)

DET = 5 (тип, имя, цена, кол-во, кнопка “Изменить”)

***Удалить заявку***

FTR = 2 (обращение к таблице заявок, обращение к внутреннему файлу)

DET = 1 (Кнопка “Удалить”)

**EO:**

RET = 2 (Запрос к локальному файлу, запрос к таблице заявок)

DET = 4 (тип, имя, кол-во, цена)

**EQ:** Запрос к бирже на получение и изменение заявок

FTR = 3 (таблица клиентов, таблица бумаг, таблица заявок)

DET = 5 (логин, тип, имя, кол-во, цена)

**Новая заявка**

Страница позволяет оформить заявку на покупку или продажу ценной бумаги. Страница состоит из 4 полей: Бумага (имя бумаги), Цена (цена по которой необходимо купить/продать бумагу), Покупка (булева переменная в значение true обозначает покупку, false – продажа) и кнопки «Оформить» - для подтверждения оформления заявки.



**ILF:**

RET = 3 (логический, строка, целое, вещественное)

DET = 3 (тип, бумага, цена, кол-во)

**EI:**

Покупка/продажа бумаги

FTR = 3 (таблица бумаг, таблица клиентов, локальный файл)

DET = 5 (бумага, цена, кол-во, тип заявки, кнопка)

**EIF:**

RET = 3 (строка, вещественный, целый, логический)

DET = 5 (бумага, цена, кол-во, тип, пользователь)

**EQ:**

FTR = 1 (таблица заявок)

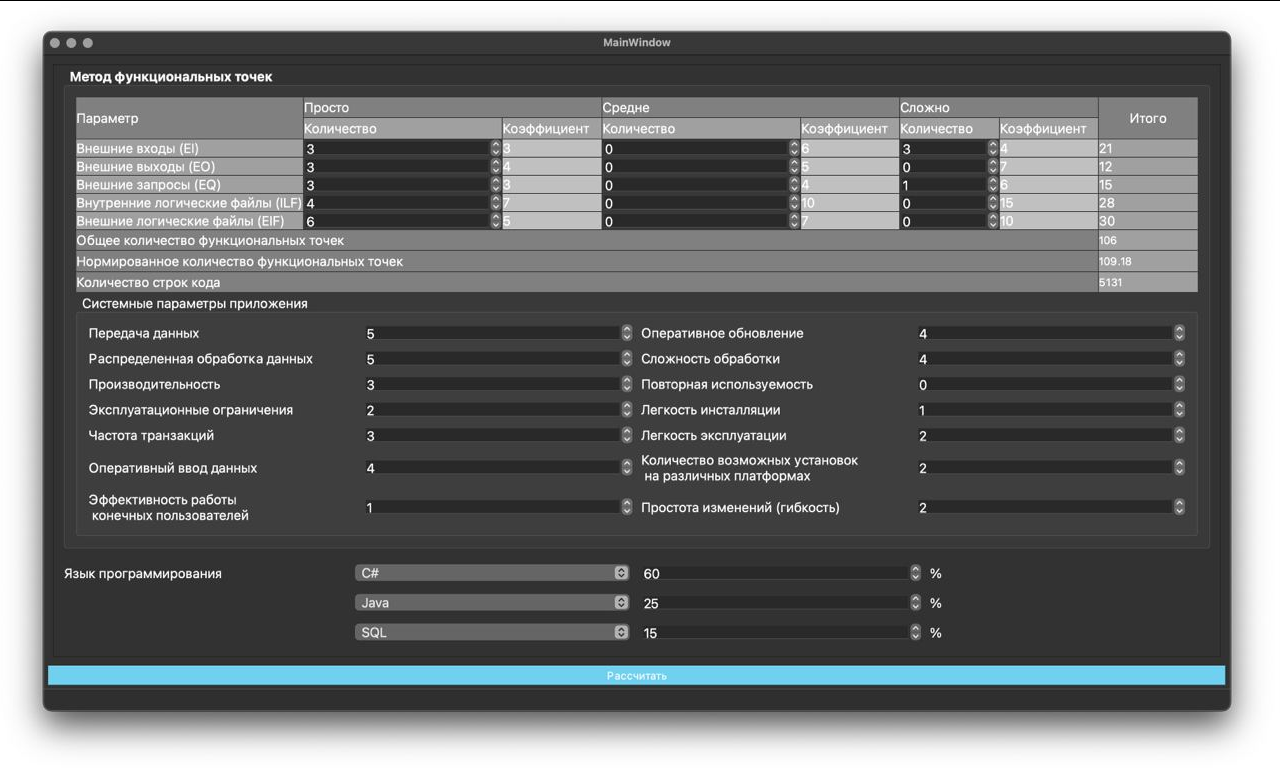
DET=5(логин, цена, тип, кол-во, бумага)

**СВОДНАЯ ТАБЛИЦА**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ILF** | | **EO** | | **EQ** | | **EI** | | **EIF** | |
| **RET** | **DET** | **FTR** | **DET** | **FTR** | **DET** | **FTR** | **DET** | **RET** | **DET** |
| **Авторизация** | 1 | 2 | 2 | 3 | 1 | 4 | 1 | 4 | 1 | 2 |
| **Биржевые сводки** | 2 | 3 | 2 | 3 | 1 | 4 | 1 | 4 | 2 | 2 |
| **Заявки** | 4 | 4 | 2 | 4 | 3 | 5 | 4 | 5 | 4 | 5 |
| 2 | 2 |
| 2 | 1 | 2 | 3 |
|
| **Новая заявка** | 3 | 3 | - | - | 1 | 5 | 3 | 5 | 3 | 5 |
| 3 | 5 |

**Итоговая таблица**

|  | **ILF** | **EO** | **EQ** | **EI** | **EIF** |
| --- | --- | --- | --- | --- | --- |
| **Авторизация** | 1 Н | 1 Н | 1 Н | 1 Н | 1 Н |
| **Биржевые сводки** | 1 Н | 1 Н | 1 Н | 1 Н | 1 Н |
| **Заявки** | 1 Н | 1 Н | 1 С | 1 Н 1 В | 3Н |
| **Новая заявка** | 1 Н | - | 1 Н | 2 В | 1 Н |



Результат:

Кол-во функциональных точек: 106

Нормированное кол-во функциональных точек: 109.18

Кол-во строк кода 5131

Методика COCOMO II

3 модели оценки стоимости в COCOMO II:

1. **Модель композиции приложения** - модель, которая подходит для проектов, созданных с помощью современных инструментальных средств. Единицей измерения служит объектная точка (учитывается количество экранов, отчётов и компонентов).
   1. рассматривается макетирование пользовательских интерфейсов
   2. оценивается производительность
   3. определяется степень зрелости технологии
2. **Модель ранней разработки архитектуры** - модель применяется для получения приблизительных оценок проектных затрат периода выполнения проекта перед тем как будет определена архитектура в целом. В качестве единиц измерения используются функциональные точки либо KSLOC.
3. **Постархитектурная модель** - наиболее детализированная модель COCOMO II, которая используется после разработки архитектуры проекта. В состав этой модели включены новые драйверы затрат, новые правила подсчёта строк кода, а также новые уравнения

|  |  |  |
| --- | --- | --- |
| **Модель композиции приложения** | **Модель ранней разработки архитектуры** | **Постархитектурная модель** |
| Грубые входные данные | Ясно понимаемы особенности проекта | Детальное описание проекта |
| Оценки низкой точности | Оценки умеренной точности | Высокоточные оценки |
| Приблизительные требования | Ясно понимаемые требования | Стабилизировавшиеся основные требования |
| Концепция архитектуры | Ясно понимаемая архитектура | Стабильная базовая архитектура |

|  |  |
| --- | --- |
| **Преимущества COCOMO II** | **Недостатки COCOMO II** |
| 1. возможен учёт многих факторов 2. универсальный метод 3. фактические данные подбираются в соответствии с реальными проектами 4. метод позволяет добавлять уникальные факторы для корректировки характеристик 5. результаты прогнозирования сопровождаются обязательной документацией 6. модель проста в освоении и применении | 1. все результаты зависят от размера программного продукта 2. игнорируются требования к характеристикам качества программного продукта 3. игнорируется изменяемость требований к программному продукту 4. игнорируются многие особенности, связанные с аппаратным обеспечением проекта |

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 системы подобного типа)
* **Гибкость процесса разработки (FLEX)** - большей часть согласованный процесс (график жесткий, точный регламентации нет)
* **Разрешение рисков в архитектуре системы (RESL)** - некоторое (40%)
* **Сплоченность команды (TEAM)** - некоторая согласованность (команда новая, но были проведены определенные мероприятия по сплочению)
* **Уровень развития процесса разработки** **(PMAT)** - начальный уровень (только начинают внедрять)

(Экспертная оценка стоимости человеко-месяца принята равной 60 000)

Расчёт **модели композиции** проекта. По страницам:

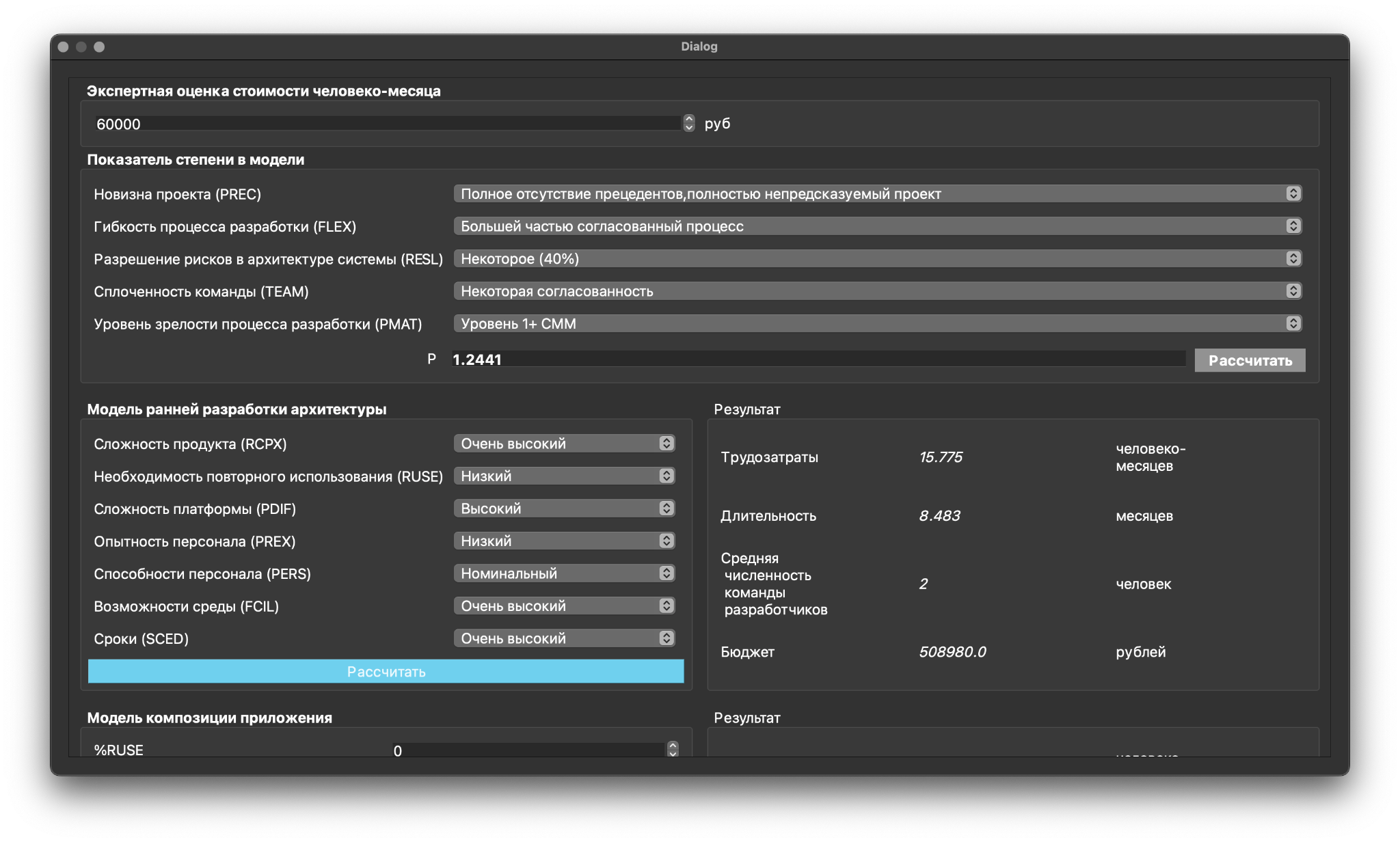
* Авторизация - 3 простых поля и 1 средней сложности, 3 поколения
* Биржевые сводки - 3 простых поля и 1 средней сложности, 3 поколения
* Заявки - 1 простое поле и 2 средней сложности, 3 поколения
* Новая заявка - 4 простых поля и 1 средней сложности, 3 поколения

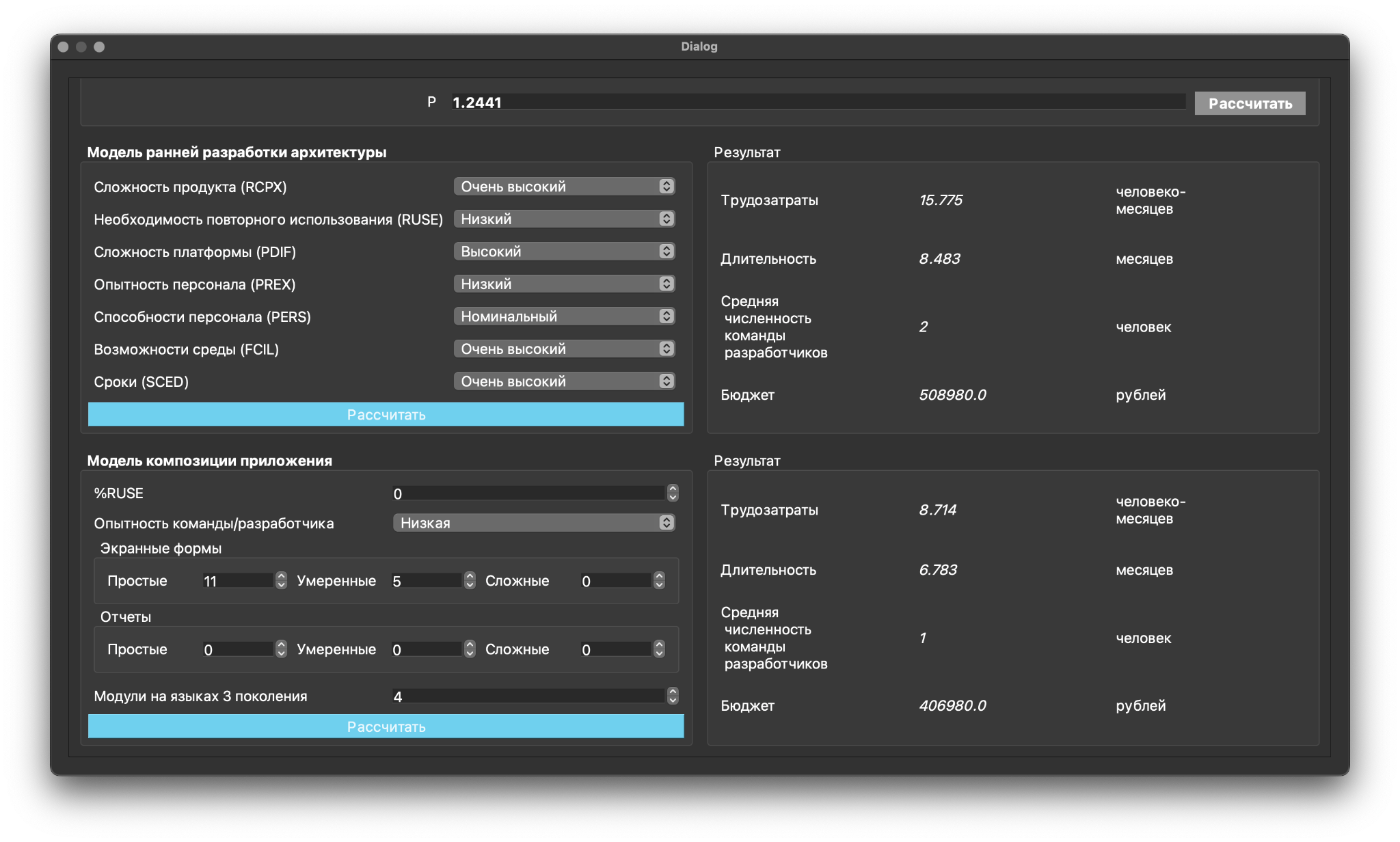
Итого:

* Полей простой сложности: **11**
* Полей средней сложности: **5**
* Полей высокой сложности: **0**
* Модулей на ЯП 3-го поколения: **4**
* Повторное использование: **не предусматривается**
* Опытность команды: **низкая**

Расчёт **модели ранней разработки** архитектуры:

* Надёжность и уровень сложности (RCPX) - **очень высокое**
* Повторное использование компонентов **не предусматривается** (RUSE)
* Возможности персонала (PERS) - **средние**
* Сложности платформы (PDIF) - **высокая**
* **Очень интенсивное** использование инструментальных средств поддержки (FCIL)
* Жесткий график (SCED)





**Вывод**

В ходе выполнения данной работы была освоена методология оценки параметров проекта COCOMO II и разработан программный инструмент для её применения. Выполнен анализ выданного задания:

* рассчитаны функциональные точки и показатель степени модели (p)
* были определены факторы, влияющие на показатель степени
* рассчитаны трудозатраты и времени по моделям:
  + ранней разработки архитектуры приложения
  + композиции приложения

По модели композиции приложения приложения прогноз более благоприятный, чем в модели ранней архитектуры приложения.

Методология COCOMO II является более сложной по сравнению с COCOMO, но позволяет более тонко настраивать параметры плана, что даёт более точный и детальный прогноз.