



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 Сусликов Д.В. Склифасовский Д.О. Платонова О.С.

Группа ИУ7-85Б

Преподаватель Барышникова М.Ю., Силантьева А.В.

Москва, 2022 г.

Задание (варианта №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 4 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

Рис.1 – Метод функциональных точек

Характеристика	Количество	Уровень сложности	Итого
Внешние вводы (EI)	4	Низкий (3)	12
Внешние выводы (EO)	2	Низкий (4)	8
Внешние запросы (EQ)	1	Низкий (3)	3
Внутренние логические файлы (ILF)	3	Низкий (7)	21
Внешние интерфейсные файлы (EIF)	1	Низкий (5)	5
Итого:			49

Рассчитать

Результаты

Нормирующий фактор:	1.03
Нормированное количество функциональных точек:	50.47
Количество функциональных точек:	49
Количество строк исходного кода:	3219

Системные параметры приложения

Передача данных	5
Распределенная обработка данных	5
Производительность	3
Эксплуатационные ограничения	2
Частота транзакций	3
Оперативный ввод данных	4
Эффективность работы конечных пользователей	1
Оперативное обновление	4
Сложность обработки	4
Повторное использование	0
Легкость установки	1
Легкость эксплуатации	2
Количество возможных установок на платформах	2
Простота изменений	2

Языки разработки

ASM	0
C	0
Cobol	0
Fortran	0
Pascal	0
C++	0
Java	25
C#	60
Ada 95	0
SQL	15
Visual C++	0
Delphi Pascal	0
Perl	0
Prolog	0

Рис.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 (команда 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:

$$p = 1.2317$$

Факторы показателя степени модели

Новизна проекта (PREC)	Почти полное отсутствие прецедентов, в знач.
Гибкость процесса разработки (FLEX)	Большой частью согласованный процесс
Анализ архитектуры системы и рисков (RESL)	Некоторое (40 %)
Сплоченность команды (TEAM)	Некоторая согласованность
Уровень развития процесса разработки (PMAT)	Уровень 1+ CMM

Рассчитать

Коэффициент P: 1.2317

Рис.2 – Факторы показателя степени модели

Модель композиции приложения

- Страница авторизации – 3 простых поля и 1 средней сложности (обращение к БД)
- Страница биржевых сводок - 3 простых поля и 1 средней сложности (обращение к БД)
- Страница заявок - 1 простое поле и 2 средней сложности (обращение к БД)
- Страница новой заявки - 4 простых поля и 1 средней сложности (обращение к БД)

Итого:

Простые поля = 11

Средней сложности = 5

Также имеются 2 модуля, написанные на ЯП третьего поколения.

Повторное использование = 0%

Опытность команды – низкая

Результат работы программы представлен на Рисунке 3.

Модель композиции приложения

	Экранные формы	Отчеты
Простые	11	0
Средние	5	2
Сложные	0	0
Модулей на языках 3 поколения	2	
RUSE	0	
Опытность команды/разработчика	Низкая	

Рассчитать

Трудозатраты 7

Время 6

Бюджет 525000

Рис.3 – Модель композиции приложения

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 = 4 (из метода функциональных точек)

Результаты расчетов представлены на Рисунке 4.

Модель ранней разработки архитектуры

PERS	Номинальный ▾
RCPX	Очень высокий ▾
RUSE	Низкий ▾
PDIF	Высокий ▾
PREX	Низкий ▾
FCIL	Очень высокий ▾
SCED	Очень высокий ▾

Рассчитать

Трудозатраты	22
Время	10
Бюджет	1650000

Рис.4 – Модель ранней разработки архитектуры

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:

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

В итоге было выяснено, что модель композиции приложения дает более оптимистичный прогноз, по сравнению с моделью ранней архитектуры приложения.