



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»

на тему: «Оценка параметров программного проекта с использованием
метода функциональных точек и модели СОСОМО II»

Вариант № 1

Студент группы ИУ7И-84Б

(Подпись, дата)

Динь Вьет Ань
(Фамилия И.О.)

Студент группы ИУ7И-84Б

(Подпись, дата)

Бу Миь Куанг
(Фамилия И.О.)

Студент группы ИУ7И-84Б

(Подпись, дата)

Каримзай А. Х.
(Фамилия И.О.)

Преподаватель

(Подпись, дата)

М.Ю. Барышникова
(И. О. Фамилия)

2024 г.

1. Задание

1.1 Цель работы

Целью лабораторной работы является продолжение знакомства с существующими методиками предварительной оценки параметров программного проекта и практическая оценка затрат по модели СОСОМО II.

1.2 Задание

1. Ознакомиться с прилагаемым к лабораторной работе теоретическим материалом (презентациями к лекциям №7 и №8).
2. На основе своего варианта задания рассчитать количество функциональных точек для разрабатываемого программного приложения. С этой целью разработать программный инструмент (Калькулятор).
3. Произвести оценку трудозатрат и длительности разработки по методике СОСОМО II с использованием моделей композиции приложения и ранней разработки архитектуры.
4. Определить среднюю численность команды разработчиков.
5. На основе экспертной оценки стоимости человеко-месяца произвести предварительную оценку бюджета проекта.
6. Дать заключение о применимости метода функциональных точек и модели СОСОМО II, а также их сравнение с промежуточной моделью СОСОМО для решения поставленной задачи с учетом своего варианта.

2. Методика функциональных точек

Данный метод используется для измерения производительности взамен устаревшего линейного подхода, где производительность измерялась количеством строк программного кода. Преимуществом данного метода является то, что поскольку применение функциональных точек основано на

изучении требований, то оценка необходимых трудозатрат может быть выполнена на самых ранних стадиях работы над проектом.

Определение числа функциональных точек является методом количественной оценки ПО, применяемым для измерения функциональных характеристик процессов его разработки и сопровождения независимо от технологии, использованной для его реализации.

Трудоемкость вычисляется на основе функциональности разрабатываемой системы, которая, в свою очередь, определяется путем выявления функциональных типов —логических групп взаимосвязанных данных, используемых и поддерживаемых приложением, а также элементарных процессов, связанных с вводом и выводом информации.

3. СОСОМО II

СОСОМО II является развитием стандартного СОСОМО. В методику входят три различные модели оценки стоимости.

3.1 Модель композиции приложения

Эта модель, которая подходит для проектов, созданных с помощью современных инструментальных средств. Единицей измерения служит объектная точка.

3.2 Модель ранней разработки архитектуры

Применяется для получения приблизительных оценок проектных затрат периода выполнения проекта перед тем как будет определена архитектура в целом. В этом случае используется небольшой набор новых драйверов затрат и новых уравнений оценки. В качестве единиц измерения используются функциональные точки либо KSLOC.

3.3 Постархитектурная модель

Наиболее детализированная модель СОСОМО II, которая используется после разработки архитектуры проекта. В состав этой модели включены новые драйверы затрат, новые правила подсчета строк кода, а также новые уравнения.

4. Проект

Компания получила заказ на разработку автоматизированной информационной системы оплаты штрафов ГИБДД. Оплата штрафов возможна через веб-интерфейс веб-портала и через приложение для мобильного телефона.

4.1 Роли пользователей

В системе предусмотрено два вида пользователей: Пользователь (User) и Администратор (Administrator). Пользователь может просматривать и оплачивать штрафы, Администратору доступен просмотр всех выплат, просмотр данных пользователей, их редактирование и создание новых пользователей.

Все записи типа «Пользователь» в системе имеют следующие поля: id, логин, пароль, тип, регистрационный номер водительского удостоверения, номер банковской карты.

4.2 Модули

Информационная система состоит из следующих модулей:

- приложение для мобильного телефона;
- веб-портал;
- модуль регистрации и авторизации;
- модуль обмена данными с системой ГИБДД;
- модуль проведения платежных транзакций.

4.2.1 Приложение для мобильного телефона

Содержимое:

- страница регистрации, где пользователь заполняет следующие поля: логин, пароль, номер водительского удостоверения, номер банковской карты;

- страницу просмотра штрафов, на которой отображаются неоплаченные штрафы, и есть кнопка “оплатить” для каждого штрафа. После нажатия кнопки пользователю приходит ответ о положительном или отрицательном результате выполнения операции.

4.2.2 Веб-портал

Обеспечивает те же функциональные возможности для Пользователя и в дополнение к этому он имеет панель Администратора.

4.2.3 Модуль регистрации и авторизации

Позволяет добавлять пользователей в базу данных.

4.2.4 Модуль обмена данными с системой ГИБДД

Предназначен для получения списка штрафов, а также оповещения ГИБДД об оплате штрафа. При отправке сообщения с номером водительского удостоверения информационной системе ГИБДД модуль обмена данными получает список штрафов.

При этом каждый штраф описывается следующими полями: номер постановления, дата постановления, имя, фамилия, отчество нарушителя, сумма штрафа. На сообщение об оплате система ГИБДД присылает подтверждение об удалении штрафа из списка неоплаченных или отказ.

Примечание: Система не хранит штрафы в своей базе данных, а получает их из системы ГИБДД при каждом запросе.

4.2.5 Модуль проведения платежных транзакций

Модуль по защищенному протоколу отправляет платежной системе запрос с указанием номера карты пользователя, номера счета ГИБДД и суммы на выполнение проведения оплаты.

Система отправляет положительный или отрицательный ответ о результате выполнения. В отличие от штрафов все операции по оплате сохраняются в базу данных.

4.3 Характеристики команды, продукта и проекта

- обмен данными - 5;
- распределенная обработка — 5;
- производительность — 3;
- эксплуатационные ограничения по аппаратным ресурсам — 0;
- транзакционная нагрузка — 3;
- интенсивность взаимодействия с пользователем (оперативный ввод данных) — 2;
- эргономические характеристики, влияющие на эффективность работы конечных пользователей — 0;
- оперативное обновление — 4;
- сложность обработки — 4;
- повторное использование — 3;
- легкость инсталляции — 0;
- легкость эксплуатации/администрирования — 3;
- портируемость — 5;
- гибкость — 0.

При разработке ПО 30% кода будет написано на SQL, 10% — на JavaScript, 60% — на Java.

К разработке проекта планируется привлечь довольно слаженную команду высокопрофессиональных разработчиков, у которых, однако, практически отсутствует опыт в разработке систем подобного типа. При этом заказчик настаивает на довольно строгом процессе с периодической демонстрацией рабочих продуктов, соответствующих этапам жизненного цикла.

Учитывая новизну проекта для команды, на этапе его подготовки был осуществлен относительно детальный анализ рисков, свойственных архитектуре разрабатываемой системы. Организация находится чуть выше второго уровня зрелости процессов разработки.

Надежность и уровень сложности (RCPX) разрабатываемой системы оцениваются как очень высокие, проект не предусматривает специальных усилий на повторное использование компонентов (RUSE). Возможности персонала (PERS) можно охарактеризовать как очень высокие, однако опыт членов команды в данной сфере (PREX) является скорее низким. Сложность платформы (PDIF) средняя. Разработка предусматривает интенсивное использование инструментальных средств поддержки (FCIL). Учитывая новизну проекта для команды и высокие требования по надежности, Заказчик не настаивает на жестком графике (SCED).

5 Лабораторная работа

5.1 Показатели проекта

В данной работе рассматриваются модель композиции приложения и модель ранней разработки архитектуры.

Показатели проекта:

- новизна проекта (PREC) — почти полное отсутствие прецедентов, в значительной мере непредсказуемый проект;
- гибкость процесса разработки (FLEX) — точный, строгий процесс разработки;
- разрешение рисков в архитектуре системы (RESL) — в целом (75%)
- сплоченность команды (TEAM) — взаимодействие как в едином целом;
- уровень развития процесса разработки (PMAT) — уровень 2.

5.2 Расчет по методу функциональных точек

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

DET — количество связанных с каждым функциональным типом элементарных 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.

Вычисление EI

— Регистрация

FTR = 1 (один внутренний логический файл);

DET = 4 (логин, пароль, номер водительского удостоверения и номер банковской карты).

— Оплата штрафа

FTR = 1 (один внутренний логический файл);

DET = 1 (дата оплаты).

— Добавление пользователя администратором;

FTR = 1 (один внутренний логический файл);

DET = 5 (логин, пароль, тип, номер карты, номер удостоверения).

- Редактирование пользователя администратором;
FTR = 1 (один внутренний логический файл);
DET = 5 (логин, пароль, тип, номер карты, номер удостоверения).
Уровень сложности – низкий.

Вычисление ЕО

- Вывод списка штрафов
FTR = 1 (один внутренний логический файл);
DET = 4 (номер, ФИО, дата, сумма).
- Вывод сообщения о статусе оплаты
FTR = 1 (один внутренний логический файл);
DET = 1 (статус оплаты).
- Вывод информации о пользователе
FTR = 1 (один внутренний логический файл);
DET = 6 (все поля из базы).
Уровень сложности – низкий.

Вычисление EQ

- Запрос на авторизацию
FTR = 1 (один внутренний логический файл);
DET = 3 (логин, пароль, флажок).
- Запрос на вывод списка
FTR = 1 (один внутренний логический файл);
DET = 4 (номер, ФИО, дата, сумма).
Уровень сложности – низкий.

Вычисление ILF

- ILF
RET = 3 (элементы записи);
DET = 6 (элементы данных).
Уровень сложности – низкий.

Вычисление EIF

— Веб

RET = 3 (элементы записи);

DET = 10 (элементы данных).

— Мобильное приложение

RET = 3 (элементы записи);

DET = 5 (элементы данных).

Уровень сложности — низкий.

5.4 Результат расчетов

5.4.1 Метод функциональных точек

Нормированное количество функциональных точек: 47.

Количество функциональных точек: 47.

Количество строк исходного кода: 3590.

The screenshot shows the COSIMO2 software interface for calculating functional points. The window title is "COSIMO2". The main menu is "Метод функциональных точек". The interface is divided into several sections:

- Left Panel:** A list of input parameters with spinners for their values.
 - Передачи данных: 5
 - Распределенная обработка данных: 5
 - Производительность: 3
 - Эксплуатационные ограничения: 0
 - Частота транзакций: 3
 - Оперативный ввод данных: 2
 - Эффективность работы конечных пользователей: 0
 - Оперативное обновление: 4
 - Сложность обработки: 4
 - Повторное использование: 3
 - Легкость установки: 0
 - Легкость эксплуатации: 3
 - Портативность: 5
 - Простота изменений: 0
- Center Panel:** A table of calculated values.

	Количество	Сложность	Итого
Внешние вводы (EI)	4	Низкий (3)	12
Внешние выходы (EO)	3	Низкий (4)	12
Внешние запросы (EQ)	2	Низкий (3)	6
Внутренние логические файлы (ILF)	1	Низкий (7)	7
Внешние интерфейсные файлы (EIF)	2	Низкий (5)	10
Всего			47

Below the table, there is a "Рассчитать" button and the following results:

 - Нормированное количество функциональных точек: 47.94
 - Количество функциональных точек: 47
 - Количество строк исходного кода: 3590
- Right Panel:** A table of language usage percentages.

Процент использования языков	Процент
ASM	0
C	0
Cobol	0
Fortran	0
Pascal	0
C++	0
Java	60
C#	0
Ada 95	0
SQL	30
Visual C++	0
Delphi Pascal	0
Perl	0
JavaScript	10

Рисунок 1 – Расчет проекта по методу функциональных точек

5.4.2 Оценка по модели COSOMO II

$p = 1:1964$

COCOMO2

Метод функциональных точек Модели композиции и ранней разработки архитектуры

Факторы показателя степени модели

Новизна проекта (PREC) Почти полное отсутствие прецедентов, в значительной мере непредсказуемый проект

Гибкость процесса разработки (FLEX) Точный, строгий процесс разработки

Анализ архитектуры системы и рисков (RESL) В целом (75 %)

Сплоченность команды (TEAM) Высокая согласованность

Уровень развития процесса разработки (PMAT) Уровень 2 CMM

R равно: **1.1964000000000001** Рассчитать Средняя зарплата: 100000

Модель ранней разработки архитектуры

PERS Очень высокий

RCPX Очень высокий

RUSE Низкий

PDFI Номинальный

PREX Низкий

FCIL Очень высокий

SCED Низкий

Рассчитать

Трудозатраты, чел/мес 13

Время, мес 8

Бюджет 1300000.0

Модель композиции приложения

Экраны формы

Простые 3

Средние 1

Сложные 0

Модулей на языках 3 поколения 3

%RUSE 0

Опытность команды/разработчика Низкая

Рассчитать

Отчеты

Простые 2

Средние 1

Сложные 0

Трудозатраты, чел/мес 6

Время, мес 6

Бюджет 600000.0

Рисунок 2 – Расчет проекта по методу композиции и ранней разработки архитектуры

5.4.3 Композиция приложения

Формы:

- страница регистрации — простая (обращение к БД);
- страница авторизации — простая (обращение к БД);
- страница штрафов — средняя (обращение к БД ГИБДД);
- страница информации о пользователе — простая (запрос к БД).

Отчеты:

- отчет о пользователях — простой (обращение к БД);
- отчет о штрафах — средний (обращение к БД ГИБДД);
- отчет о платежах — простой (обращение к БД).

Итого:

- простые формы — 3;
- средние формы — 1;

- простые отчеты — 2;
- средние отчеты — 1;
- модули на ЯП третьего поколения — 3;
- повторное использование — 0%;
- опыт команды — очень низкий;

5.4.4 Модель ранней разработки архитектуры

- PERS (возможности персонала) – очень высокий;
- RCPX (надежность и уровень сложности разрабатываемой системы) – очень высокий;
- RUSE (повторное использование компонентов) – низкий;
- PDIF (сложность платформы разработки) – номинальный;
- PREX (опыт персонал) – низкий;
- FCIL (средства поддержки) – очень высокий;
- SCED (график работ) – низкий;
- KSLOC = 3.6 (из метода функциональных точек).

6. Выводы

Методика СОСОМО II, как и ее первая версия, позволяет досочно быстро оценить длительность и трудозатраты проекта, основываясь на субъективных данных. В условиях отсутствия объективной информации о предполагаемых трудозатратах особенно важно правильно спрогнозировать характеристики проекта.

Методика функциональных точек позволяет оценить размер программного продукта на этапе его проектирования. С помощью этого подхода можно применять методики СОСОМО, которым необходимо знание о размере продукта. Хотя методика функциональных точек является неточной (как и любая методика прогнозирования), этих результатов достаточно для общего понимания объемов работ и получения приблизительных оценок параметров проекта.

В ходе выполнения лабораторной работы было выяснено, что модель композиции приложения дает менее оптимистичный прогноз, по сравнению с моделью ранней архитектуры приложения.