



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 ИУ7-83Б
(Группа)

(Подпись, дата)

Марченко В.
(И. О. Фамилия)

Студент ИУ7-83Б
(Группа)

(Подпись, дата)

Науменко А. А.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Барышникова М. Ю.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Силантьева А. В.
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

1	Лабораторная работа	3
1.1	Цель работы	3
1.2	Методика СОСОМО II	3
1.3	Постановка задачи	4
1.4	Метод функциональных точек	6
1.5	Модель композиции приложения	9
1.6	Модель ранней архитектуры приложения	12
1.7	Вывод	13

1 Лабораторная работа

1.1 Цель работы

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

1.2 Методика COSOMO II

Проект COSOMO II стартовал в 1995 году в центре по разработке ПО USC при финансовой и технической поддержке большого количества промышленных предприятий, таких как AT&T, BELL Labs, Hewlett-Packard, Rational, Texas Instruments, Lockheed Martin, Motorola, Xerox и др.

Проект имел триединую задачу:

- 1) разработать модель для оценки стоимости и сроков создания ПО для того жизненного цикла, который будет применяться в конце 20 — начале 21 века;
- 2) разработать базу данных стоимости программных проектов и осуществить инструментальную поддержку методов усовершенствования модели стоимости;
- 3) создать количественную аналитическую схему для оценки технологий создания ПО и их экономического эффекта.

Существует три модели оценки стоимости COSOMO II.

Модель композиции приложения — это модель, которая подходит для проектов, созданных с помощью современных инструментальных средств. Единицей измерения служит объектная точка.

Модель ранней разработки архитектуры. Эта модель 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.

Постархитектурная модель — наиболее детализированная модель СО-СОМО II, которая используется после разработки архитектуры проекта. В состав этой модели включены новые драйверы затрат, новые правила подсчета строк кода, а также новые уравнения.

1.3 Постановка задачи

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. 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.

Информационная система состоит из следующих модулей:

- 1) приложение для мобильного телефона;
- 2) веб-портал;
- 3) модуль регистрации и авторизации;
- 4) модуль обмена данными с системой ГИБДД;
- 5) модуль проведения платежных транзакций.

Приложение для мобильного телефона имеет: страницу 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.

Веб-портал способен обеспечивать те же функциональные возможности для Пользователя и в дополнение к этому он имеет панель Администратора.

Модуль регистрации и авторизации позволяет добавлять пользователей в базу данных.

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.

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.

Характеристики команды, продукта и проекта.

Характеристики продукта:

- 1) обмен данными — 5;
- 2) распределенная обработка — 5;
- 3) производительность — 3;
- 4) эксплуатационные ограничения по аппаратным ресурсам — 0;
- 5) транзакционная нагрузка — 3;
- 6) интенсивность взаимодействия с пользователем (оперативный ввод данных) — 2;
- 7) эргономические характеристики, влияющие на эффективность работы конечных пользователей — 0;
- 8) оперативное обновление — 4;

- 9) сложность обработки — 4;
- 10) повторное использование — 3;
- 11) легкость инсталляции — 0;
- 12) легкость эксплуатации/администрирования — 3;
- 13) портируемость — 5;
- 14) гибкость — 0.

При разработке ПО 30% кода будет 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).

1.4 Метод функциональных точек

Определение границ программного обеспечения.

ПО состоит из 5-и модулей (по условию). Из них 2 используются для обмена данными со сторонними системами — система ГИБДД и система проведения платежей.

Внутренние логические файлы.

Единственный ILF-файл используется для хранения данных о пользователях. В каждой строке хранится идентификатор, логин, пароль, тип, номер в/у и номер банковской карты пользователя. Значения всех атрибутов, кроме идентификатора, лучше хранить в виде строк. Идентификатор может быть как строкой, так и целым числом. Пусть он будет целым числом, тогда для данного файла $RET = 2$, а $DET = 6$. Уровень сложности этого файла — низкий.

Внешние интерфейсные файлы.

Первый EIF-файл используется для хранения данных о штрафах в системе ГИБДД. В каждой строке хранится номер постановления, дата постановления, имя, фамилия, отчество, сумма штрафа. По условию, поиск информации о штрафе происходит по номеру в/у, поэтому будем считать, что этот атрибут тоже хранится в данном файле. Номер постановления — целое число, дата постановления — тип «дата», сумма штрафа — тип «деньги», остальные атрибуты — строки. Тогда для данного файла $RET = 4$, а $DET = 7$. Уровень сложности этого файла — низкий.

Второй EIF-файл используется для хранения данных о транзакциях в системе проведения платежей. В каждой строке хранится номер карты пользователя, номер счета ГИБДД и сумма на выполнение проведения оплаты. Сумма — тип «деньги», остальные атрибуты — строки. Тогда для данного файла $RET = 2$, а $DET = 3$. Уровень сложности этого файла — низкий.

Внешние вводы.

Регистрация пользователя. Пользователь вводит логин, пароль, номер в/у, номер банковской карты, нажимает кнопку «Зарегистрироваться» и получает подтверждение регистрации. Данная операция ссылается на один ILF. Тогда $DET = 6$, $FTR = 1$. Уровень сложности — низкий.

Оплата штрафа. Пользователь только нажимает на кнопку «Оплатить», сам ничего не вводит, но получает уведомление о статусе операции. Данная операция ссылается на один EIF. Тогда $DET = 2$, $FTR = 1$. Уровень сложности — низкий.

Редактирование информации о пользователе (для администратора). Администратор может поменять идентификатор, логин, пароль, тип, номер в/у, номер банковской карты, нажимает кнопку «Изменить» и получает

подтверждение. Данная операция ссылается на один ILF. Тогда $DET = 8$, $FTR = 1$. Уровень сложности — низкий.

Создание пользователя (для администратора). Администратор вводит идентификатор, логин, пароль, тип, номер в/у, номер банковской карты, нажимает кнопку «Создать» и получает подтверждение. Данная операция ссылается на один ILF. Тогда $DET = 8$, $FTR = 1$. Уровень сложности — низкий.

Внешние запросы.

Авторизация пользователя. Пользователь вводит логин, пароль, нажимает кнопку «Войти» и получает подтверждение авторизации. Данная операция ссылается на один ILF. Тогда $DET = 4$, $FTR = 1$. Уровень сложности — низкий.

Просмотр штрафов. Пользователь только нажимает на кнопку «История штрафов», сам ничего не вводит, но получает уведомление о статусе операции. Данная операция ссылается на один EIF. Тогда $DET = 2$, $FTR = 1$. Уровень сложности — низкий.

Оплата штрафа. Происходит через стороннюю систему, поэтому рассматривается как EI и EQ. Данная операция ссылается на один EIF. Тогда $DET = 2$, $FTR = 1$. Уровень сложности — низкий.

Информация о выплатах (для администратора). Администратор только нажимает на кнопку «История выплат», сам ничего не вводит, но получает уведомление о статусе операции. Данная операция ссылается на один EIF. Тогда $DET = 2$, $FTR = 1$. Уровень сложности — низкий.

Информация о пользователе (для администратора). Администратор вводит номер в/у или логин и нажимает на кнопку «Посмотреть профиль», получает уведомление о статусе операции. Данная операция ссылается на один ILF. Тогда $DET = 4$, $FTR = 1$. Уровень сложности — низкий.

Внешние выводы.

Ошибка при создании/редактировании пользователя с уже существующим логином, номером в/у или идентификатором. $DET = 2$ (повторяющееся поле и сообщение об ошибке), $FTR = 1$. Уровень сложности — низкий.

Ошибка при авторизации пользователя с несуществующим логином или неверным паролем. $DET = 1$ (сообщение об ошибке), $FTR = 1$. Уровень сложности — низкий.

Ошибка при оплате штрафа. DET = 1 (сообщение об ошибке), FTR = 1.
Уровень сложности — низкий.

Итог

Важно отметить, что по условию есть два приложения — веб и мобильное. Поэтому некоторые EI, EO и EQ продублируем.

Низких ILF — 1, средних ILF — 0, высоких ILF — 0, низких EIF — 2, средних EIF — 0, высоких EIF — 0, низких EI — 6, средних EI — 0, высоких EI — 0, низких EO — 6, средних EO — 0, высоких EO — 0, низких EQ — 7, средних EQ — 0, высоких EQ — 0.

	Низкий	Средний	Высокий
Внешние вводы	6	0	0
Внешние выводы	6	0	0
Внешние запросы	7	0	0
Внутренние логические файлы	1	0	0
Внешние интерфейсные файлы	2	0	0

Рисунок 1.1 – Количество функциональных точек

Пересчитаем функциональные точки в KSLOC:

$$KSLOC = 0.3 \cdot 81.6 \cdot 13 + 0.1 \cdot 81.6 \cdot 56 + 0.6 \cdot 81.6 \cdot 53 = 3.37. \quad (1.1)$$

Полученной оценкой воспользуемся позже.

1.5 Модель композиции приложения

Подсчитаем объектные точки.

Формы:

- 1) регистрация (мобильное приложение) — умеренная сложность;
- 2) авторизация (мобильное приложение) — низкая сложность;
- 3) регистрация (веб-портал) — умеренная сложность;
- 4) авторизация (веб-портал) — низкая сложность;
- 5) редактирование информации о пользователе для администратора (веб-портал) — умеренная сложность.

Отчеты:

- 1) информация о штрафе (мобильное приложение) — умеренная сложность;
- 2) информация о штрафе (веб-портал) — умеренная сложность;
- 3) информация о выплатах для администратора (веб-портал) — умеренная сложность;
- 4) информация о пользователе для администратора (веб-портал) — низкая сложность.

Модули:

- 1) приложение для мобильного телефона;
- 2) веб-портал;
- 3) модуль регистрации и авторизации;
- 4) модуль обмена данными с системой ГИБДД;
- 5) модуль проведения платежных транзакций.

Итого: форм низкой сложности — 2, форм умеренной сложности — 3, форм высокой сложности — 0, отчетов низкой сложности — 1, отчетов умеренной сложности — 3, отчетов высокой сложности — 0, модулей на языках третьего поколения — 5.

По правилу подсчета объектных точек получим:

$$OP = 1 \cdot 2 + 2 \cdot 3 + 3 \cdot 0 + 2 \cdot 1 + 5 \cdot 3 + 8 \cdot 0 + 10 \cdot 5 = 2 + 6 + 2 + 15 + 50 = 75. \quad (1.2)$$

Скорректируем полученное значение:

$$NOP = 75 \cdot \frac{(100 - 0)}{100} = 75. \quad (1.3)$$

Трудозатраты считаются по формуле:

$$LC = \frac{NOP}{PROD} = \frac{75}{7} = 10.71, \quad (1.4)$$

где $PROD = 7$ (т. к. опытность команды по условию низкая).

Время найдем из следующей формулы:

$$T = 3 \cdot LC^{0.33+0.2 \cdot (p-1.01)}, \quad (1.5)$$

где p — показатель степени, вычисляемый на основе пяти показателей.

С помощью определенных параметров посчитаем трудозатраты, затраты и бюджет (при средней з/п 60000 руб/мес.), используя реализованное ПО.

The screenshot shows the COSIMO II software interface with the following sections:

- Project Parameters (Left Panel):**
 - PREC (новизна проекта): Почти полное отсутствие прецедентов, в
 - FLEX (гибкость процесса разработки): Некоторые послабления в процессе
 - RESL (разрешение рисков в архитектуре системы): Почти полное (90%)
 - TEAM (сплоченность команды): Высокая согласованность
 - PMAT (уровень зрелости процесса разработки): Уровень 3 CMM
 - RCPX (надежность и сложность продукта): Очень высокий
 - RUSE (повторное использование компонентов): Очень низкий
 - PDIF (сложность платформы): Номинальный
 - PERS (способности персонала): Очень высокий
 - PREX (опытность персонала): Низкий
 - FCIL (возможности среды): Очень высокий
 - SCED (сроки): Низкий
- Input Fields (Bottom Left):**
 - Процент повторного использования кода программы, %: 0
 - Опытность команды/разработчика: Низкая
 - Простые экранные формы, шт.: 2
 - Средние экранные формы, шт.: 3
 - Сложные экранные формы, шт.: 0
 - Простые отчеты, шт.: 1
 - Средние отчеты, шт.: 3
 - Сложные отчеты, шт.: 0
 - Кол-во модулей на языках 3-го поколения, шт.: 5
- Results (Right Panel):**
 - Количество строк, тыс.: 3,37
 - Средняя з/п, тыс. руб.: 60
 - Модель ранней разработки архитектуры:**
 - Трудозатраты, чел.-мес.: 6,85
 - Время, мес.: 5,97
 - Бюджет, руб.: 411052,61
 - Кол-во работников, чел.: 2
 - Модель композиции приложения:**
 - Трудозатраты, чел.-мес.: 10,71
 - Время, мес.: 7
 - Бюджет, руб.: 642857,14
 - Кол-во работников, чел.: 2
- Buttons (Bottom Right):**
 - Рассчитать
 - Функциональные точки

Рисунок 1.2 – Время, трудозатраты и бюджет для модели композиции приложения

Трудозатраты — 10.71 чел.-мес., время — 7 мес., бюджет — 642 857.14

руб, количество работников — 2 человека.

1.6 Модель ранней архитектуры приложения

При подсчете функциональных точек было получено оценочное значение количества строк исходного кода в тысячах — 3.37.

Для вычисления трудозатрат и времени используются следующие формулы:

$$LC = 2.45 \cdot eArch \cdot KSLOC^p, \quad (1.6)$$

$$T = 3 \cdot LC^{0.33+0.2 \cdot (p-0.1)}. \quad (1.7)$$

Множитель $eArch$ вычисляется на основе семи показателей.

Project Characteristics	
PREC (новизна проекта)	Почти полное отсутствие прецедентов, в
FLEX (гибкость процесса разработки)	Некоторые послабления в процессе
RESL (разрешение рисков в архитектуре системы)	Почти полное (90%)
TEAM (сплоченность команды)	Высокая согласованность
PMAT (уровень зрелости процесса разработки)	Уровень 3 CMM

Product Characteristics	
RCPX (надежность и сложность продукта)	Очень высокий
RUSE (повторное использование компонентов)	Очень низкий
PDIF (сложность платформы)	Номинальный
PERS (способности персонала)	Очень высокий
PREX (опытность персонала)	Низкий
FCIL (возможности среды)	Очень высокий
SCED (сроки)	Низкий

User-defined Parameters	
Процент повторного использования кода программы, %	0
Опытность команды/разработчика	Низкая
Простые экранные формы, шт.	2
Средние экранные формы, шт.	3
Сложные экранные формы, шт.	0
Простые отчеты, шт.	1
Средние отчеты, шт.	3
Сложные отчеты, шт.	0
Кол-во модулей на языках 3-го поколения, шт.	5

Calculated Results	
Количество строк, тыс.	3,37
Средняя з/п, тыс. руб	60

Model of early architecture development	
Трудозатраты, чел.-мес.	6,85
Время, мес.	5,97
Бюджет, руб	411052,61
Кол-во работников, чел.	2

Model of application composition	
Трудозатраты, чел.-мес.	10,71
Время, мес.	7
Бюджет, руб	642857,14
Кол-во работников, чел.	2

Рассчитать

Функциональные точки

Рисунок 1.3 – Время, трудозатраты и бюджет для модели ранней архитектуры приложения

Трудозатраты — 6.85 чел.-мес., время — 5.97 мес., бюджет — 411 052.61 руб, количество работников — 2 человека.

1.7 Вывод

Модель СОСОМО II, как и ее предшественница, позволяет достаточно быстро оценить длительность и трудозатраты проекта, основываясь на субъективных данных.

Метод функциональных точек позволяет оценить размер программного продукта на этапе его проектирования. Оценив размер ПО, можно применять методики СОСОМО, которым необходимо знание о размере продукта.

И хотя метод функциональных точек является неточным (как и любой другой метод прогнозирования), этих результатов достаточно для общего понимания объема работ и получения приблизительных оценок параметров проекта.