



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 дисциплине
«Экономика программной инженерии»

Тема Оценка параметров программного проекта с использованием МФТ и СОСОМО II

Студенты Борисов М. А., Леонов В. В., Трошкин Н. Р.

Группа ИУ7-86Б

Вариант 1

Оценка (баллы) _____

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

Цель работы

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

Содержание проекта

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.

Постановка задачи

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.

Модули приложения

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 отсылает 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 включают в себя:

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. Портруемость — 5.
14. 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

Модель композиции приложения

Для расчета трудоемкости проекта с использованием модели композиции приложения подсчитаем количество объектных точек.

Модули — по заданию имеем 5 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.

Экранных форм всего 5 — 2 для мобильного приложения и 3 для веб-портала. Панели регистрации в мобильной и веб-версии считаем простым изображением (1 точка), страницу просмотра штрафов в обеих версиях — изображением умеренной сложности (2 точки), а панель администратора — сложным (3 точки). Итого 2 простых, 2 средних и 1 сложное изображение.

Также имеется 4 отчета — о положительном или отрицательном результате выполнения транзакции (простой, 2 точки), об удалении штрафа из списка неоплаченных (2 точки), о неоплаченных штрафах (умеренной сложности, 5 точек) и отчет администратора (сложный, 8 точек). Всего 2 простых, и по 1 среднему и сложному.

Всего объектных точек:

$$(5 \cdot 10) + (2 + 2 \cdot 2 + 1 \cdot 3) + (2 \cdot 2 + 5 + 8) = 76 \quad (1)$$

Допускаем, что проект абсолютно новый и процент переиспользуемого кода = 0%.

Так как разработчики имеют очень высокую квалификацию, но опыт работы в данной предметной области низкий, уровень опытности и возможности разработчиков примем за номинальный.

Возможности среды разработки считаем высокими.

Модель ранней разработки архитектуры

Для расчета трудоемкости проекта с использованием модели ранней разработки архитектуры подсчитаем количество функциональных точек. Данная модель используется, когда нужно получить приблизительные затраты на выполнение проекта, прежде чем будет определена общая архитектура продукта. Функциональные точки как единицы измерения можно перевести в количество строк кода в зависимости от того, на каком языке планируется написать этот код.

Метод функциональных точек основан на изучении требований к проекту, в отличие от линейного подхода, где объем работ измерялся на основе оценки количества строк программного кода. В данном методе оценку можно выполнить на самых ранних стадиях работы над проектом, в частности на этапе ранней разработки архитектуры. Трудоемкость вычисляется на основе функциональности системы, определяемой на основе логических групп взаимосвязанных данных и элементарных процессов ввода-вывода.

Рассмотрим каждый модуль.

Модуль регистрации и авторизации:.

- Внутренний логический файл (ILF): таблица с пользователями. Содержит 6 полей — id (целое число), логин (строка), пароль (строка), тип (целое число), регистрационный номер водительского удостоверения (строка), номер банковской карты (строка). DET = 6, RET = 2 — низкая сложность (**ранг 7**).
- Внешний ввод (EI): Форма регистрации из 4 полей — логин, пароль,

номер водительского удостоверения, номер банковской карты, все поля имеют строковый тип. DET = 4, FTR = 1 (таблица в БД с информацией о пользователях) — низкая сложность (**ранг 3**).

- Внешний ввод (EI): Форма авторизации из 2 полей — логин и пароль (оба значения строковые). DET = 2, FTR = 1 (таблица в БД с информацией о пользователях) — низкая сложность (**ранг 3**).

Модуль обмена данными с системой ГИБДД:

- Внешний интерфейсный файл (EIF): файл в системе ГИБДД с информацией о штрафах, состоит из 6 полей: номер постановления (строка), дата постановления (дата), имя (строка), фамилия (строка), отчество нарушителя (строка), сумма штрафа (вещественное число). RET = 3, DET = 6 — низкая сложность (**ранг 5**).
- Внешний ввод (EI): получить информации о штрафах — номер удостоверения (строка), взаимодействует с внешним интерфейсным файлом (системой ГИБДД) с информацией о штрафах. DET = 1, FTR = 1 — низкая сложность (**ранг 3**).
- Внешний вывод (EO): информация о штрафе, поля соответствуют внешнему файлу, DET = 6, FTR = 1 (внешняя система ГИБДД) — низкая сложность (**ранг 4**).
- Внешний вывод (EO): отчет об удалении штрафа из списка неоплаченных, состоит из строкового статуса (DET = 1, FTR = 0) — низкая сложность (**ранг 4**).
- Внешний запрос (EQ): запрос к системе ГИБДД о текущих неоплаченных штрафах пользователя, DET = 7 (на вход системе 1 поле, полученное из EI, а получаем 6 полей, которые выводятся в EO), FTR = 1 (система ГИБДД) — низкая сложность (**ранг 3**).
- Внешний запрос (EQ): запрос к системе ГИБДД о статусе оплаченного штрафа, DET = 2 (номер постановления на входе и статус на выходе), FTR = 1 (система ГИБДД) — низкая сложность (**ранг 3**).

Модуль проведения платежных транзакций:

- Внутренний логический файл (ILF): таблица в БД с информацией о транзакциях. Хранятся номер карты пользователя (строка), номер счета получателя (строка), дата транзакции (дата), сумма транзакции (вещественное число), номер постановления штрафа (строка). RET = 3, DET = 5 — низкая сложность (**ранг 7**).
- Внешний ввод (EI): оплата транзакции — передается номер постановления (строка), сумма штрафа (целое число), ID пользователя (целое число), взаимодействует с внутренним логическим файлом с информацией о пользователях. DET = 3, FTR = 1 — низкая сложность (**ранг 3**).
- Внешний запрос (EQ): запрос к платежной системе с указанием номера карты, номера счета ГИБДД и суммы штрафа, взаимодействие происходит с файлом о пользователях, чтобы узнать номер карты пользователя, и с файлом о транзакциях, чтобы сохранить в него информацию о транзакции (DET = 4, FTR = 2) — низкая сложность (**ранг 3**).
- Внешний вывод (EO): отчет об успешности транзакции, состоит из строкового статуса (DET = 1, FTR = 0) — низкая сложность (**ранг 4**).

Модуль администратора:

- Внешний запрос (EQ): получить информацию из БД о транзакциях — FTR = 1, DET = 5 — низкая сложность (**ранг 3**).
- Внешний ввод (EI): создание/редактирование пользователя — тот же, что и при регистрации.
- Внешний запрос (EQ): просмотреть данные пользователей, информацию из БД о пользователях — FTR = 1, DET = 6 — низкая сложность (**ранг 3**).

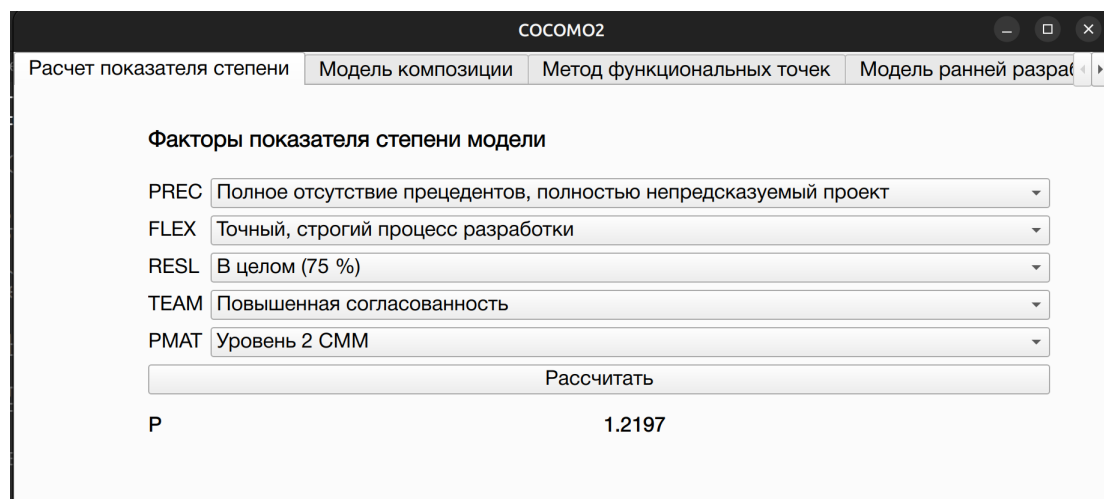
Расчет показателя степени

Из данных, приведенных в постановке задачи, были выбраны следующие параметры для расчета показателя степени в модели COSOMO II:

- $PREC$ = Полное отсутствие прецедентов, полностью непредсказуемый проект — 6.2
- $FLEX$ = Точный, строгий процесс разработки — 5.07
- $RESL$ = В целом (75 %) — 2.83
- $TEAM$ = Повышенная согласованность — 2.19
- $PMAT$ = Уровень 2 CMM — 4.68

1 Результаты расчетов

Сперва вычислим показатель степени для расчета времени работ. Это требуется в обеих моделях. Интерфейс приложения и результат расчета приведены на рисунке 1.



The screenshot shows the COSOMO2 application window. It has a title bar with standard window controls. Below the title bar is a tabbed interface with four tabs: 'Расчет показателя степени' (selected), 'Модель композиции', 'Метод функциональных точек', and 'Модель ранней разраб'. The main area under the selected tab is titled 'Факторы показателя степени модели'. It contains five dropdown menus for the factors: PREC (selected: 'Полное отсутствие прецедентов, полностью непредсказуемый проект'), FLEX (selected: 'Точный, строгий процесс разработки'), RESL (selected: 'В целом (75 %)'), TEAM (selected: 'Повышенная согласованность'), and PMAT (selected: 'Уровень 2 CMM'). Below these is a 'Рассчитать' (Calculate) button. At the bottom, the result is displayed as 'Р' followed by the value '1.2197'.

Рисунок 1: Расчет показателя степени

Таким образом, значение показателя степени равно примерно 1.22.

На рисунке 2 приведены характеристики проекта, полученные при помощи модели композиции.

В результате имеем 5.69 человекомесечев, 5.73 месяца и бюджет, равный 569 000 рублей, при средней зарплате в 100 000 рублей. Получается, что численность команды в среднем равна 1 человеку.

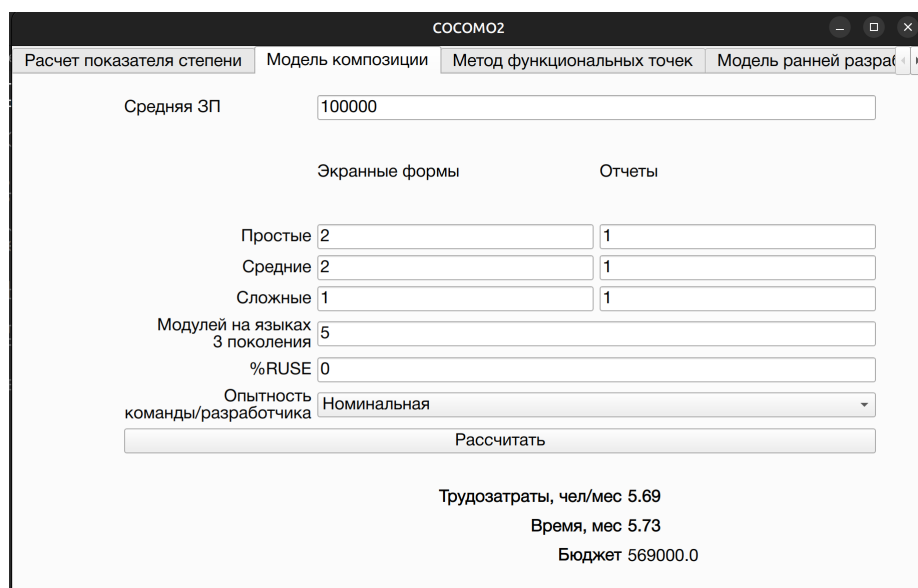


Рисунок 2: Модель композиции

На рисунке 3 показан расчет количества функциональных точек и прогноз количества строк кода на их основе. Количество функциональных точек подсчитано по данным, указанным выше.

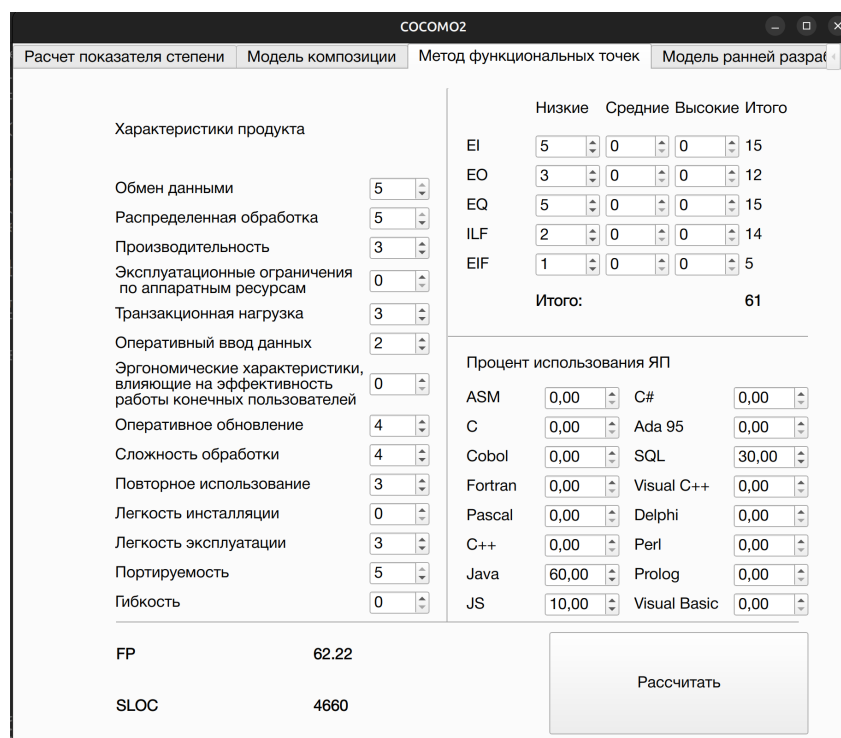


Рисунок 3: Расчет функциональных точек

Получаем 61 функциональную точку, но с учетом корректирующих параметров, указанных в левой части экрана, получаем 62.22 функциональных точки, которые конвертируются примерно в 4660 строк кода.

И наконец, на рисунке 4 приведены характеристики проекта, подсчитанные по модели ранней разработки архитектуры.

СОСОМО2

степени | Модель композиции | Метод функциональных точек | Модель ранней разработки архитектуры

Средняя ЗП 100000

PERS Очень высокий

RCPX Очень высокий

RUSE Низкий

PDIF Номинальный

PREX Низкий

FCIL Высокий

SCED Низкий

Рассчитать

Трудозатраты, чм 22.15
Время, мес 9.5
Бюджет 2215000.0

Рисунок 4: Модель ранней разработки архитектуры

Согласно расчетам, трудозатраты составляют 22.15 человекомесяцев, время на разработку выросло по сравнению с предыдущей моделью до 9.5 месяцев, а бюджет при той же средней зарплате равен 2 215 000 рублей.

Средняя численность команды составляет примерно 2.33 человека. Так как модель ранней разработки архитектуры является более точной, данные цифры можно считать более реалистичными, чем полученные по модели композиции приложения.

Сравним с результатами, которые были получены для данного проекта с использованием методики СОСОМО и приведены на рисунке 5.

Получаем, что трудозатраты составляют 16.96 человекомесяцев, время на разработку = 9.68 месяцев, а бюджет при той же средней зарплате равен почти 1,7 млн рублей. То есть порядок трудозатрат, времени и бюджета близок к тому, что показывает модель ранней разработки архитектуры.

Constructive cost model | Борисов Максим

Атрибуты программного продукта

RELY (Требуемая надежность)

Очень высокий

DATA (Размер базы данных)

Номинальный

CPLX (Сложность продукта)

Очень высокий

Атрибуты компьютера

TIME (Ограничение времени выполнения)

Номинальный

STOR (Ограничение объема основной памяти)

Номинальный

VIRT (Изменчивость виртуальной машины)

Номинальный

TURN (Время реакции компьютера)

Номинальный

Атрибуты персонала

ACAP (Способности аналитика)

Очень высокий

AEXP (Знание приложений)

Номинальный

PCAP (Способности программиста)

Очень высокий

VEXP (Знание виртуальной машины)

Номинальный

LEXR (Знание языка программирования)

Номинальный

Атрибуты проекта

MODP (Использование современных методов)

Номинальный

TOOL (Использование программных инструментов)

Высокий

SCED (Требуемые сроки разработки)

Очень высокий

Размер проекта

KLOC (Количество строк)

5

Рассчет

Распределение работ и времени по стадиям жизненного цикла

Вид деятельности	Работа (%)	Работа (чел.-м)	Время (%)	Время (месяцы)
Планирование и определение требований	8	1,26	36	2,56
Проектирование продукта	18	2,83	36	2,56
Детальное проектирование	25	3,93	18	1,28
Кодирование и тестирование отдельных модулей	26	4,08	18	1,28
Интеграция и тестирование	31	4,87	28	1,99
Итого	100	15,7	100	7,12
Итого с планированием	108	16,96	136	9,68

Распределение работ по видам деятельности

Вид деятельности	Бюджет (%)	Человеко-месяцы
Анализ требований	4	0,63
Проектирование продукта	12	1,88
Программирование	44	6,91
Тестирование	6	0,94
Верификация и аттестация	14	2,2
Канцелярия проекта	7	1,1
Управление конфигурацией и обеспечение качест...	7	1,1
Создание руководства	6	0,94
Итого	100	15,7

Рисунок 5: Результат расчетов по модели COSOMO

Выводы

Методика COSOMO II, как и ее первая версия, позволяет досрочно 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 COSOMO, 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 более оптимистичный прогноз, по сравнению с моделью ранней архитектуры приложения.

12