



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»

ОТЧЁТ

по лабораторной работам № 6

по курсу «Экономика программной инженерии»

«Предварительная оценка параметров программного проекта»

Студент: Керимов А. Ш.

Группа: ИУ7-84Б

Вариант: 4

Преподаватель: Барышникова М. Ю.

Москва.
2021 г.

Краткое описание методики СОСОМО

COConstructive COst MOdel — конструктивная модель стоимости.

$$\text{Трудозатраты} = C1 \cdot EAF \cdot (\text{Размер})^{p1}$$

$$\text{Время} = C2 \cdot (\text{Трудозатраты})^{p2}$$

- + Трудозатраты (работа) — количество человеко-месяцев;
- + C1 — масштабирующий коэффициент;
- + EAF — уточняющий фактор, характеризующий предметную область, персонал, среду и инструментарий, используемый для создания рабочих продуктов процесса;
- + Размер — размер конечного продукта (кода, созданного человеком), измеряемый в исходных инструкциях (DSI, delivered source instructions), которые необходимы для реализации требуемой функциональной возможности;
- + p1 — показатель степени, характеризующий экономию при больших масштабах, присущую тому процессу, который используется для создания конечного продукта; в частности, способность процесса избегать непроизводительных видов деятельности (доработок, бюрократических проволочек, накладных расходов на взаимодействие);
- + Время — общее количество месяцев;
- + C2 — масштабирующий коэффициент для сроков исполнения
- + p2 — показатель степени, который характеризует инерцию и распараллеливание, присущие управлению разработкой ПО.

Таблица 1 — Режимы модели СОСОМО

| Название режима | Размер проекта | Описание | Среда разработки | C1 | p1 | C2 | p2 |
|-----------------|----------------|--|--|-----|------|-----|------|
| Обычный | До 50 KLOC | Некрупный проект разрабатывается небольшой командой, для которой нехарактерны нововведения, разработчики знакомы с инструментами и языком программирования | Стабильная | 3,2 | 1,05 | 2,5 | 0,38 |
| Промежуточный | 50—500 KLOC | Относительно небольшая команда занимается проектом среднего размера, в процессе разработки | Среда характеризуется незначительной нестабильностью | 2,0 | 1,12 | 2,5 | 0,35 |

| | | | | | | | |
|------------|----------------|---|---|-----|-----|-----|------|
| | | необходимы определенные инновации | | | | | |
| Встроенный | Более 500 KLOC | Большая команда разработчиков трудится над крупным проектом, необходим значительный объем инноваций | Среда состоит из множества нестабильных элементов | 2,8 | 1,2 | 2,5 | 0,32 |

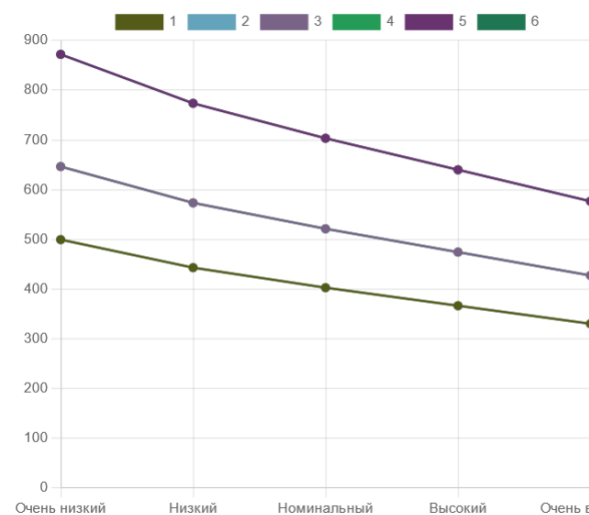
Таблица 2 — Значение драйверов затрат в модели COCOMO

| Идентификатор | Уточняющий фактор работ | Очень низкий | Низкий | Номинальный | Высокий | Очень высокий |
|---------------------------------------|--|--------------|--------|-------------|---------|---------------|
| <i>Атрибуты программного продукта</i> | | | | | | |
| RELY | Требуемая надёжность | 0,75 | 0,86 | 1,0 | 1,15 | 1,4 |
| DATA | Размер базы данных | | 0,94 | 1,0 | 1,08 | 1,16 |
| CPLX | Сложность продукта | 0,7 | 0,85 | 1,0 | 1,15 | 1,3 |
| <i>Атрибуты компьютера</i> | | | | | | |
| TIME | Ограничение времени выполнения | | | 1,0 | 1,11 | 1,5 |
| STOP | Ограничение объёма основной памяти | | | 1,0 | 1,06 | 1,21 |
| VIRT | Изменчивость виртуальной машины | | 0,87 | 1,0 | 1,15 | 1,30 |
| TURN | Время реакции компьютера | | 0,87 | 1,0 | 1,07 | 1,21 |
| <i>Атрибуты персонала</i> | | | | | | |
| ACAP | Способности аналитика | 1,46 | 1,19 | 1,0 | 0,86 | 1,30 |
| AEXP | Знание приложений | 1,29 | 1,15 | 1,0 | 0,91 | 1,15 |
| PCAP | Способности программиста | 1,42 | 1,17 | 1,0 | 0,86 | 0,71 |
| VEXP | Знание виртуальной машины | 1,21 | 1,1 | 1,0 | 0,9 | 0,82 |
| LEXP | Знание языка программирования | 1,14 | 1,07 | 1,0 | 0,95 | 0,7 |
| <i>Атрибуты проекта</i> | | | | | | |
| MODP | Использование современных методов | 1,24 | 1,1 | 1,0 | 0,91 | 0,82 |
| TOOL | Использование программных инструментов | 1,24 | 1,1 | 1,0 | 0,91 | 0,82 |
| SCED | Требуемые навыки разработки | 1,23 | 1,08 | 1,0 | 1,04 | 1,1 |

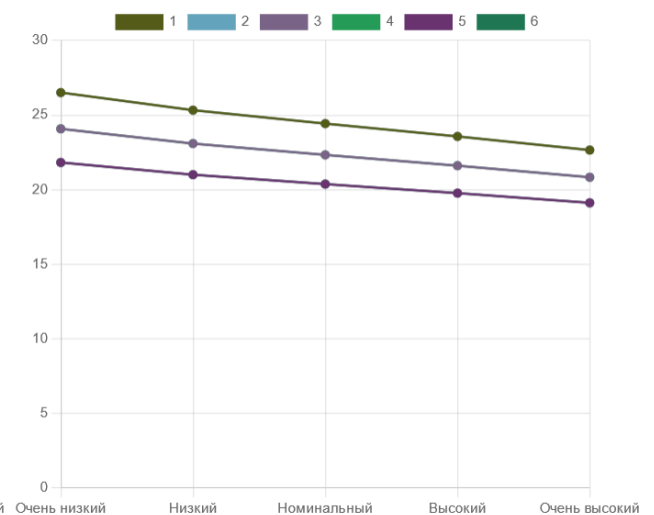
Задание 1

Исследовать влияние уровня автоматизации процесса разработки на трудоёмкость (PM) и время разработки проекта (TM) для модели COCOMO и разных типов проектов (обычного, промежуточного, встроенного). Получить значения PM и TM по всем типам проектов для одного и того же значения параметра размера программного кода (SIZE), выбрав номинальный, низкий и высокий уровень использования современных методов и программных инструментов. Результаты исследований оформить графически.

PM трудоемкость проекта



TM время разработки проекта



Графики 1.* — изменяя MODP, 100 KLOC, Обычный вариант

Графики 2.* — изменяя TOOL, 100 KLOC, Обычный вариант

Графики 3.* — изменяя MODP, 100 KLOC, Промежуточный вариант

Графики 4.* — изменяя TOOL, 100 KLOC, Промежуточный вариант

Графики 5.* — изменяя MODP, 100 KLOC, Встроенный вариант

Графики 6.* — изменяя TOOL, 100 KLOC, Встроенный вариант

На рисунке графики MODP и TOOL попарно совпадают для каждого варианта проекта. Видно, что уровень автоматизации проекта положительно сказывается на трудоёмкости и времени разработки.

Задание 2

Компания получила заказ на разработку программного обеспечения для рабочей станции дизайнера автомобиля. Заказчик следующим образом определил проблемную область в своей спецификации: ПО должно формировать 2-х и 3-х мерные изображения для дизайнера, система должна иметь стандартизированный графический интерфейс, геометрические и прикладные данные должны содержаться в базе данных (планируемый размер базы данных не более 200 тыс. записей). При анализе проекта его размер был предварительно оценён в 140 000 строк кода. Проект реализуется по промежуточному варианту. Все показатели драйверов затрат, кроме трёх имеют номинальное значение. Знание языка программирования имеет высокую оценку, использование современных методов — очень высокую оценку и использование программных инструментов — низкую, так как используется стандартная среда визуального программирования. Произвести оценку проекта по методике COCOMO.

Атрибуты программного продукта

| | | |
|------|----------------------|---------------|
| RELY | Требуемая надёжность | Номинальный ▾ |
| DATA | Размер базы данных | Номинальный ▾ |
| CPLX | Сложность продукта | Номинальный ▾ |

Атрибуты проекта

| | | |
|------|--|-----------------|
| MODP | Использование современных методов | Очень высокий ▾ |
| TOOL | Использование программных инструментов | Низкий ▾ |
| SCED | Требуемые сроки разработки | Номинальный ▾ |

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 | Номинальный ▾ |
| AEXP | Знание приложений | Номинальный ▾ |
| PCAP | Способности программиста | Номинальный ▾ |
| VEXP | Знание виртуальной машины | Номинальный ▾ |
| LEXP | Знание языка программирования | Высокий ▾ |

Формула

для оценки осн. работ и сроков

Промежуточный ▾

SIZE

140

KLOC

Таблицы 1.* — PM: 651.20; TM: 24.14, MODP = 0.82, TOOL = 1.1, LEXP = 0.95, 140 KLOC, Промежуточный вариант

Таблица 1.1 — Распределение работ и времени по стадиям жизненного цикла

| Вид деятельности | Трудозатраты (чм) | Время (м) | Кол-во сотрудников (Work/Time) |
|--|-------------------|-----------|--------------------------------|
| Планирование и определение требований | 52.10 | 8.69 | 6.00 |
| Проектирование продукта | 117.22 | 8.69 | 13.00 |
| Детальное проектирование | 162.80 | 4.35 | 37.00 |
| Кодирование и тестирование отдельных модулей | 169.31 | 4.35 | 39.00 |
| Интеграция и тестирование | 201.87 | 6.76 | 30.00 |
| Итого | 703.30 | 32.83 | |

Таблица 1.2 — Предположительный бюджет

| | |
|--|--------------|
| Анализ требований (4%) | 3686056.00 |
| Проектирование продукта (12%) | 11058168.00 |
| Программирование (44%) | 40546615.00 |
| Тестирование (6%) | 5529084.00 |
| Верификация и аттестация (14%) | 12901196.00 |
| Канцелярия проекта (7%) | 64505979.00 |
| Управление конфигурацией и обеспечение качества (7%) | 6450598.00 |
| Создание руководств (6%) | 5529084.00 |
| Непредвиденные риски(+20%) | 18430280.00 |
| Итого | 110581678.00 |

Предварительная оценка бюджета может быть произведена по формуле

$$B = Pay \cdot Work,$$

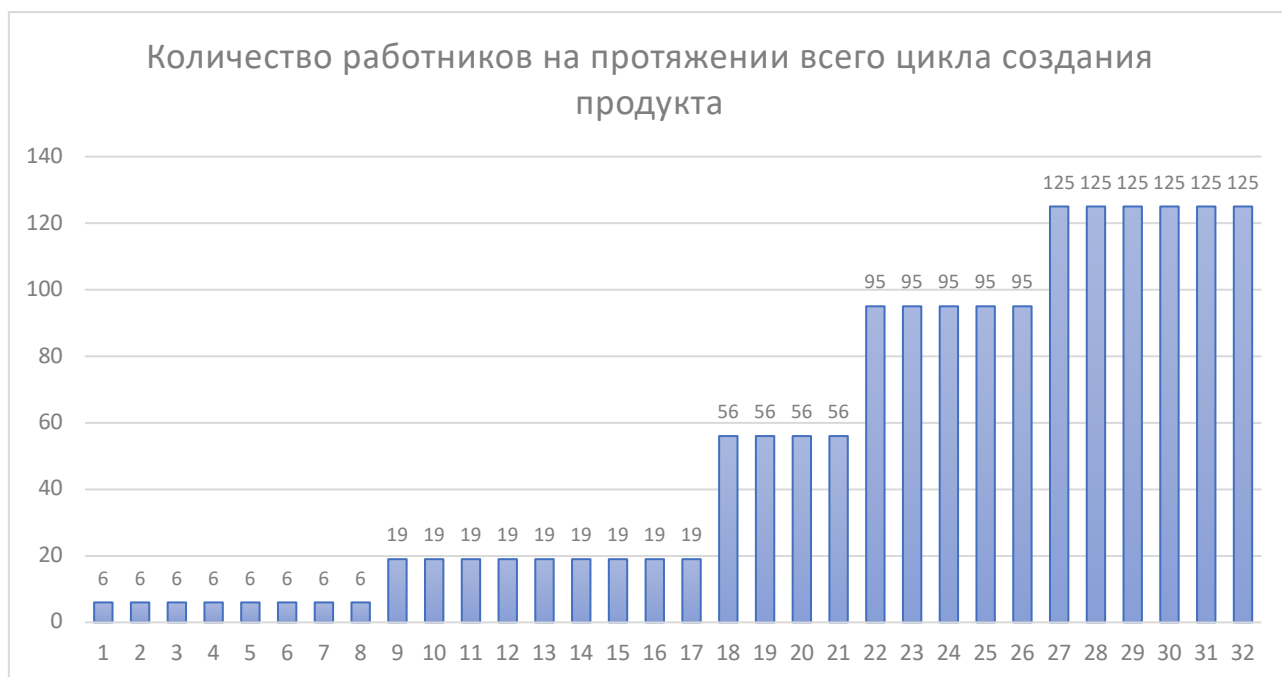
где Pay — стоимость человеко-месяца. Возьмём зарплаты специалистов с аналитики Хабр.Карьеры за 2020 год:

1. Системный аналитик — 100 000 Р.
2. Архитектор ПО — 192 000 Р.

3. Главный разработчик — 140 000 Р, продуктовый менеджер — 150 000 Р (трудоzатраты разделить поровну).

4. Разработчик — 120 000 Р, QA-инженер — 100 000 Р (трудоzатраты поровну).

5. Разработчик — 120 000 Р, QA-инженер — 100 000 Р (трудоzатраты поровну).



Вывод

Достоинства модели COSOMO:

- + Метод является достаточно универсальным и может поддерживать различные режимы и уровни программных разработок.

- + При расчётах используются множители и показатели степени, полученные на основе анализа данных большого количества практически реализованных проектов.

- + Предложенные драйверы затрат хорошо подгоняются под специфику конкретной организации.

- + Точность оценок повышается по мере накопления в организации опыта применения модели.

- + Метод снабжён обширной документацией и прост в применении.

Недостатки модели COSOMO:

- Все уровни зависят от оценки размера — точность оценки размера оказывает влияние на точность оценки трудозатрат, времени разработки, подбор персонала и оценку производительности.
- Метод основан на каскадной модели жизненного цикла и прежде всего не учитывает изменяемость требований.
- Слишком поверхностное внимание уделено вопросам обеспечения безопасности и надёжности.
- Модель не учитывает возможности повторного использования кода, итерационные возвраты по этапам жизненного цикла, объектно-ориентированные технологии разработки ПО.