

Петрозаводский государственный университет
Институт Математики и информационных технологий
Кафедра Информатики и математического обеспечения

Руководство проектом. Предварительная оценка затрат.

Лектор:

к.т.н., доцент Богоявленский Ю. А.
ybgv@cs.karelia.ru



- Оценка на основе LOC и FP метрик при планировании. Конструктивная модель стоимости. Модель композиции приложения.
- Модель раннего этапа проектирования.
- Модель этапа пост-архитектуры.
- Предварительная оценка проекта. Анализ чувствительности. Сценарий понижения зарплаты

Содержание

- Выполнение оценки проекта на основе LOC- и FP-метрик
- Шаги процесса оценки
- Конструктивная модель стоимости
- Модель композиции приложения

Выполнение оценки проекта на основе LOC- и FP-метрик

Процесс руководства программным проектом начинается с множества действий, объединяемых общим названием: планирование проекта. Первое из этих действий — выполнение оценки. Оно закладывает фундамент для других действий по планированию проекта. При оценке проекта чрезвычайно высока цена ошибок. Очень важно провести оценку с минимальным риском.

Цель этой деятельности — сформировать предварительные оценки, которые позволяют:

- предъявить заказчику корректные требования по стоимости и затратам на разработку программного продукта;
- составить план программного проекта.

Выполнение оценки проекта на основе LOC- и FP-метрик

При выполнении оценки возможны два варианта использования LOC- и FP-данных:

- в качестве оценочных переменных, определяющих размер каждого элемента продукта;
- в качестве метрик, собранных за прошлые проекты и входящих в метрический базис фирмы.

Шаги процесса оценки

Шаги 1, 2, 3

- 1 Область назначения проектируемого продукта разбивается на ряд функций, каждую из которых можно оценить индивидуально:

$$f_1, f_2, \dots, f_n.$$

- 2 Для каждой функции f_i планировщик формирует лучшую $LOC(FP)$, худшую $LOC(FP)$ и вероятную оценку $LOC(FP)$. Используются опытные данные (из метрического базиса) или интуиция. Диапазон значения оценок соответствует степени предусмотренной неопределенности.
- 3 Для каждой функции f_i в соответствии с β - распределением вычисляется ожидаемое значение LOC- (или FP-) оценки:

$$LOC = (LOC + LOC + 4 \times LOC) / 6$$

Шаги процесса оценки

Шаг 4

Определяется значение LOC- или FP- производительности разработки функции. Используется один из трех подходов:

- 1 для всех функций принимается одна и та же метрика средней производительности , взятая из метрического базиса;
- 2 для i -й функции на основе метрики средней производительности вычисляется настраиваемая величина производительности: $i = \times (LOC/LOC)$, где LOC —средняя LOC-оценка, взятая из метрического базиса (соответствует средней производительности);
- 3 для i -и функции настраиваемая величина производительности вычисляется по аналогу, взятому из метрического базиса:
 $i = \times (LOC/LOC)$.

Шаги процесса оценки

Шаг 5 и 6

5. Вычисляется общая оценка затрат на проект:

для первого подхода

$$= (\sum_{i=1}^n LOC) / [\text{чел.-мес.}];$$

Для второго и третьего подходов

$$= \sum_{i=1}^n (LOC /) [\text{чел.-мес.}];$$

6. Вычисляется общая оценка стоимости проекта:

для первого и второго подходов

$$= (\sum_{i=1}^n LOC) \times _$$

Для третьего подхода

$$= \sum_{i=1}^n (LOC \times _)$$

Конструктивная модель стоимости

Автор оригинальной модели — Барри Боэм (1981) — дал ей название COCOMO 81 (Constructive Cost Model) и ввел в ее состав три разные по сложности статистические подмодели:

- базисная COCOMO;
- промежуточная COCOMO;
- усовершенственная COCOMO.

Подмодели COCOMO 81 могут применяться к трем типам программных проектов. По терминологии Боэма их образуют:

- распространенный тип;
- полунезависимый тип;
- встроенный тип.

Конструктивная модель стоимости

В 1995 году Боэм ввел более совершенную модель COSOMO II, ориентированную на применение в программной инженерии 21 века. В состав COSOMO II входят:

- модель композиции приложения;
- модель раннего этапа проектирования;
- модель этапа пост-архитектуры.

Для описания моделей COSOMO II требуется информация о размере программного продукта. Возможно использование LOC-оценок, объектных указателей, функциональных указателей.

Модель композиции приложения

Модель композиции используется на ранней стадии разработки ПО, когда:

- рассматривается макетирование пользовательских интерфейсов;
- обсуждается взаимодействие ПО и компьютерной системы;
- оценивается производительность;
- определяется степень зрелости технологии.

Модель композиции приложения ориентирована на применение объектных указателей.

Объектный указатель — средство косвенного измерения ПО, для его расчета определяется количество экранов (как элементов пользовательского интерфейса), отчетов и компонентов, требуемых для построения приложения.

Таблица 3.15. Оценка количества объектных указателей

Тип объекта		Количество	ВЕС			Итого
			Простой	Средний	Сложный	
1	Экран	□	×1	×2	×3	- □
2	Отчет	□	×2	×5	×8	= □
3	3GL-компонент	□			×10	= □
Объектные указатели						= □

Таблица 3.16. Оценка сложности экрана

ЭКРАНЫ	Количество серверных (срв) и клиентских (клт) таблиц данных		
Количество представлений	Всего < 4 (< 2 срв, < 3 клт)	Всего < 8 (2–3 срв, 3–5 клт)	Всего ≥ 8 (> 3 срв, > 5 клт)
< 3	Простой	Простой	Средний
3–7	Простой	Средний	Сложный
> 8	Средний	Сложный	Сложный

Таблица 3.17. Оценка сложности отчета

ОТЧЕТЫ	Количество серверных (срв) и клиентских (клт) таблиц данных		
Количество представлений	Всего < 4 (< 2 срв, < 3 клт)	Всего < 8 (2–3 срв, 3–5 клт)	Всего ≥ 8 (> 3 срв, > 5 клт)
0 или 1	Простой	Простой	Средний
2 или 3	Простой	Средний	Сложный
≥ 4	Средний	Сложный	Сложный

Модель композиции приложения

Для учета реальных условий разработки вычисляется процент повторного использования программных компонентов % REUSE и определяется количество новых объектных указателей NOP:

Для оценки затрат, основанной на величине NOP, надо знать скорость разработки продукта PROD. Эту скорость определяют по таблице, учитывающей уровень опытности разработчиков и зрелость среды разработки.

Таблица 3.18. Оценка скорости разработки

Опытность/возможности разработчика	Зрелость/возможности среды разработки	PROD
Очень низкая	Очень низкая	4
Низкая	Низкая	7
Номинальная	Номинальная	13
Высокая	Высокая	25
Очень высокая	Очень высокая	50

Модель композиции приложения

Проектные затраты оцениваются по формуле:

$ЗАТРАТЫ = NOP / PROD$ [чел.-мес.], где $PROD$ — производительность разработки, выраженная в терминах объектных указателей. В более развитых моделях дополнительно учитывается множество масштабных факторов, формирователей затрат, процедур поправок.

■ Модель раннего этапа проектирования.

- ▶ Основное уравнение модели.
- ▶ Оценка показателя B .
- ▶ Оценка множителя M_e .
- ▶ Оценка слагаемого $ЗАТРАТЫ_{auto}$.

■ Модель этапа пост-архитектуры.

- ▶ Основное уравнение модели.
- ▶ Оценка коэффициента $K_{\sim req}$.
- ▶ Оценка размера проекта.
- ▶ Факторы, используемые при оценке M_p .
- ▶ Оценка стоимости и длительности проекта.

Модель раннего этапа проектирования

Основное уравнение модели имеет следующий вид:

$$ЗАТРАТЫ = A * РАЗМЕР^B * M_e + ЗАТРАТЫ_{auto}[\text{чел.-мес.}],$$

где:

- масштабный коэффициент $A = 2,5$;
- $РАЗМЕР$ – размер проекта (в тысячах LOC);
- показатель B отражает нелинейную зависимость затрат от размера проекта;
- множитель поправки M_e зависит от 7 факторов затрат, характеризующих продукт, процесс и персонал;
- слагаемое $ЗАТРАТЫ_{auto}$ отражает затраты на автоматически генерируемый программный код.

Ранний этап проектирования: оценка показателя B

Значение показателя степени B изменяется в диапазоне 1,01...1,26, зависит от пяти масштабных факторов W_i . и вычисляется по формуле:

$$B = 1,01 + 0,01 * \sum_{i=1}^5 W_i.$$

Общая характеристика масштабных факторов приведена в таблице на слайде 18. Оценки принимают 6 значений: от очень низкой (5) до сверхвысокой (0).

Ранний этап проектирования: оценка показателя B

Масштабный фактор (W_i)	Описание
Предсказуемость PREC	Отражает предыдущий опыт организации в реализации проектов этого типа.
Гибкость разработки FLEX	Отражает степень гибкости процесса разработки.
Разрешение архитектуры/риска RESL	Отражает степень выполняемого анализа риска.
Связность группы TEAM	Отражает, насколько хорошо разработчики группы знают друг друга и насколько удачно они совместно работают.
Зрелость процесса PMAT	Означает зрелость процесса в организации. Вычисление этого фактора может выполняться по вопроснику CMM

Таблица: Характеристика масштабных факторов

Ранний этап проектирования: оценка множителя M_e

Множитель поправки M_e зависит от набора формироваелей затрат, перечисленных в таблице:

Обозначение	Название
PERS	Возможности персонала
RCPX	Надежность и сложность продукта
RUSE	Требуемое повторное использование
PDIF	Трудность платформы
PREX	Опытность персонала
FCIL	Средства поддержки
SCED	График

Таблица: Формирователи затрат для раннего этапа проектирования

Ранний этап проектирования: оценка множителя M_e

Алгоритм оценки M_e :

- 1 Для каждого формирователя затрат определяется оценка (по 6-бальной шкале), где 1 соответствует очень низкому значению, а 6 — сверхвысокому значению.
- 2 На основе оценки для каждого формирователя по таблице Боэма определяется множитель затрат EM_i .
- 3 Итоговая оценка M_e определяется по следующей формуле:

$$M_e = \prod_{i=1}^7 EM_i$$

Ранний этап проектирования: оценка $ЗАТРАТЫ_{auto}$

Слагаемое $ЗАТРАТЫ_{auto}$ используется, если некоторый процент программного кода генерируется автоматически и вычисляется по следующей формуле:

$$ЗАТРАТЫ_{auto} = (KALOC * (AT/100))/ATPROD$$

, где:

- $KALOC$ – количество строк автоматически генерируемого кода (в тысячах строк);
- AT – процент автоматически генерируемого кода (от всего кода системы);
- $ATPROD$ – производительность автоматической генерации кода.

Модель этапа пост-архитектуры

Основное уравнение модели имеет следующий вид:

$$ЗАТРАТЫ = A * K_{\sim req} * РАЗМЕР^B * M_p + ЗАТРАТЫ_{auto}[\text{чел.-мес.}],$$

где:

- масштабный коэффициент $A = 2,5$;
- коэффициент $K_{\sim req}$ учитывает возможные изменения в требованиях;
- показатель B отражает нелинейную зависимость затрат от размера проекта;
- $РАЗМЕР$ – размер проекта;
- множитель поправки M_p является произведением 17 факторов затрат, каждый из которых меняется от 1 до 6.
- слагаемое $ЗАТРАТЫ_{auto}$ отражает затраты на автоматически генерируемый программный код.

Этап пост-архитектуры: оценка коэффициента $K_{\sim req}$

Изменчивость требований приводит к повторной работе, требуемой для учета предлагаемых изменений, оценка их влияния выполняется по формуле:

$$K_{\sim req} = 1 + (BRAK/100),$$

где:

- $BRAK$ процент кода, отброшенного (модифицированного) из-за изменения требований.

Этап пост-архитектуры: оценка размера проекта

Размер проекта и продукта определяют по формуле:

$$PA3MEP = PA3MEP_{new} + PA3MEP_{reuse},$$

где:

- $PA3MEP_{new}$ — размер нового программного кода;
- $PA3MEP_{reuse}$ — размер повторно используемого программного кода, определяется по формуле:

$$PA3MEP_{reuse} = KASLOC * \frac{100 - AT}{100} * \frac{AA + SU + 0.4DM + 0.3CM + 0.3IM}{100},$$

где:

- $KASLOC$ — количество строк повторно используемого кода, который должен быть модифицирован (в тысячах строк);

Этап пост-архитектуры: оценка размера проекта

- AT — процент автоматически генерируемого кода;
- DM — процент модифицируемых проектных моделей;
- CM — процент модифицируемого программного кода;
- IM — процент затрат на интеграцию, требуемых для подключения повторно используемого ПО;
- SU фактор, основанный на стоимости понимания добавляемого ПО; изменяется от 50 (для сложного неструктурированного кода) до 10 (для хорошо написанного объектно-ориентированного кода);
- AA — фактор, который отражает стоимость решения о том, может ли ПО быть повторно используемым; зависит от размера требуемого тестирования и оценивания (величина изменяется от 0 до 8).

Факторы, использующиеся при оценке M_p

■ Факторы продукта:

- ▶ требуемая надежность ПО – *RELY*;
- ▶ размер базы данных – *DATA*;
- ▶ сложность продукта – *CPLX*;
- ▶ документирование требований жизненного цикла – *DOCU*.

■ Факторы платформы (виртуальной машины):

- ▶ ограничения времени выполнения – *TIME*;
- ▶ ограничения оперативной памяти – *STOR*;
- ▶ изменчивость платформы – *PVOL*.

■ Факторы персонала:

- ▶ возможности аналитика – *ACAP*;
- ▶ возможности программиста – *PCAP*;
- ▶ опыт работы с приложением – *AEXP*;
- ▶ опыт работы с платформой – *PEXP*;
- ▶ опыт работы с языком и утилитами – *LTEX*;
- ▶ непрерывность персонала – *PCON*.

■ Факторы проекта:

- ▶ использование программных утилит – *TOOL*;
- ▶ мультисетевая разработка – *SITE*;
- ▶ требуемый график разработки – *SCED*.

Оценка стоимости и длительности проекта

Стоимость проекта оценивается по формуле:

$$\text{СТОИМОСТЬ} = \text{ЗАТРАТЫ} * \text{РАБ_КОЭФ},$$

где:

- *ЗАТРАТЫ* – ранее рассчитанные затраты на проект;
- *РАБ_КОЭФ* – стоимость одного человека-месяца.

Длительность проекта (в календарных днях) оценивается по формуле:

$$TDEV = 3 * \text{ЗАТРАТЫ}^{0,33+0,2(B-1,01)} * SCEDPercentage/100[\text{мес.}],$$

где:

- *B* – ранее рассчитанный показатель степени;
- *SCEDPercentage* – процент увеличения (уменьшения) номинального графика.

Предварительная оценка проекта. Анализ чувствительности. Сценарий понижения зарплаты.

- Предварительная оценка проекта.
- Проверка расчетов с помощью FR- указателей.
- Анализ чувствительности.
- Сценарий понижения зарплаты.

Предварительная оценка проекта. Пример

Предварительную оценку проекта будем производить с применением методики оценки на основе LOC- и FP-метрик

Пример: поступил заказ от концерна «СУПЕРАВТО».

Задача: создать ПО для рабочей станции дизайнера автомобиля (РДА).

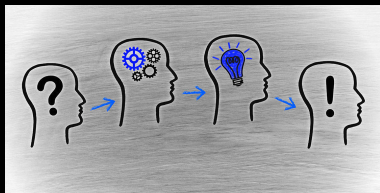
Заказчик определил предметную область проекта в своей спецификации

Предварительная оценка проекта. Предметная область проекта

- ПО РДА должно формировать 2- и 3-мерные изображения для дизайнера;
- Дизайнер должен вести диалог с РДА и управлять им с помощью стандартизованного графического пользовательского интерфейса;
- Геометрические и прикладные данные должны содержаться в базе данных РДА;
- Модули проектного анализа рабочей станции должны формировать данные для широкого класса дисплеев SVGA;
- ПО РДА должно управлять и вести диалог с периферийными устройствами (мышь, дигитайзер, плоттер, сканер, струйный и лазерный принтеры).

Порядок действий

- Прежде всего надо детализировать предметную область. Следует выделить базовые функции ПО и очертить количественные границы. Очевидно, нужно определить, что такое «стандартизованный графический пользовательский интерфейс», каким должен быть размер и другие характеристики базы данных РДА и т. д.
- Идентифицировать основные функции ПО.



Предварительная оценка проекта. Основные функции ПО

- 1 Средства управления пользовательским интерфейсом (СУПИ).
- 2 Анализ 2-мерной графики (А2Г).
- 3 Анализ 3-мерной графики (А3Г).
- 4 Управление базой данных (УБД).
- 5 Средства компьютерной дисплейной графики (КДГ).
- 6 Управление периферией (УП).
- 7 Модули проектного анализа (МПА).

Предварительная оценка проекта. Количественная оценка функций

- Каждой функции эксперты предоставляют лучшее, худшее и вероятное значения.
- Ожидаемая LOC-оценка реализации функции:

$$LOC_{ожі} = \frac{(LOC_{лучші} + LOC_{худші} + 4LOC_{вероятні})}{6},$$

Функция	Лучш. [LOC]	Вероят. [LOC]	Худш. [LOC]	Ожид. [LOC]	Уд. стоимость [\$ / LOC]	Стоимость [\$]	Произв. [LOC / чел.-мес.]	Затраты [чел.-мес.]
СУПИ	1800	2400	2650	2340				
А2Г	4100	5200	7400	5380				
А3Г	4600	6900	8600	6800				
УБД	2950	3400	3600	3350				
КДГ	4050	4900	6200	4950				
УП	2000	2100	2450	2140				
МПА	6600	8500	9800	8400				
Итого				33 360				

Определение удельной стоимости и производительности

В архиве фирмы, хранятся данные метрического базиса, собранные по уже выполненным проектам.

Пусть из метрического базиса извлечены данные по функциям-аналогам, представленные в таблице.

Функция	LOC _{анл}	УД_СТОИМОСТЬ _{анл} [\$ / LOC]	ПРОИЗВ _{анл} [LOC / чел.-мес.]
СУШИ	585	14	1260
А_Г	3000	20	440
УБД	1117	18	720
КДГ	2475	22	400
УП	214	28	1400
МПА	1400	18	1800

Определение удельной стоимости и производительности

Из таблицы видно, что

- наибольшая удельная стоимость — строка функции управления периферией,
- наименьшая удельная стоимость — строка функции управления пользовательским интерфейсом.

Считается, что удельная стоимость строки - константа, не изменяется от реализации к реализации.

Стоимость разработки каждой функции:

$$\text{СТОИМОСТЬ}_i = \text{LOC}_{\text{ож}i} * \text{УД_СТОИМОСТЬ}_{\text{ан}i}$$

Вычисление производительности и затрат на разработку каждой функции

- Для вычисления производительности разработки каждой функции используется подход настраиваемой производительности:

$$\text{ПРОИЗВ}_i = \text{ПРОИЗВ}_{\text{ані}} * \left(\frac{LOC_{\text{ані}}}{LOC_{\text{ожі}}_i} \right)$$

- Выражение для определения затрат на разработку каждой функции:

$$\text{ЗАТРАТЫ}_i = \left(\frac{LOC_{\text{ожі}}_i}{\text{ПРОИЗВ}_i} \right) [\text{чел.-мес.}]$$

Предварительная оценка проекта. Конечная таблица оценки проекта

Все необходимые для завершения расчетов данные получены и занесены в таблицу.

Функция	Лучш.	Вероят.	Худш.	Ожид. [LOC]	Уд. стои- мость [\$/LOC]	Стои- мость [\$]	Произв. [LOC/ чел.- мес.]	За- траты [чел.- мес.]
СУПИ	1800	2400	2650	2340	14	32 760	315	7,4
A2Г	4100	5200	7400	5380	20	107 600	245	21,9
A3Г	4600	6900	8600	6800	20	136 000	194	35,0
УБД	2950	3400	3600	3350	18	60 300	240	13,9
КДГ	4050	4900	6200	4950	22	108 900	200	24,7
УП	2000	2100	2450	2140	28	59 920	140	15,2
МПА	6600	8500	9800	8400	18	151 200	300	28,0
Итого				33 360		\$656 680		146

Проверка расчетов с помощью FP- указателей: Таблица оценки информационных характеристик проекта

Пусть все информационные характеристики имеют средний уровень сложности.

Результаты экспертной оценки:

Характеристика		Лучш.	Вероят.	Худш.	Ожид.	Слож-ность	Количество
1	Вводы	20	24	30	24	×4	96
2	Выводы	12	15	22	16	×5	80
3	Запросы	16	22	28	22	×4	88
4	Логические файлы	4	4	5	4	×10	40
5	Интерфейсные файлы	2	2	3	2	×7	14
Общее количество							318

Проверка расчетов с помощью FR- указателей: Таблица оценки системных параметров проекта

Коэффициенты регулировки сложности		Оценка
F_1	Передачи данных	2
F_2	Распределенная обработка данных	0
F_3	Производительность	4
F_4	Распространенность используемой конфигурации	3
F_5	Скорость транзакций	4
F_6	Оперативный ввод данных	5
F_7	Эффективность работы конечного пользователя	5
F_8	Оперативное обновление	3
F_9	Сложность обработки	5
F_{10}	Повторная используемость	4
F_{11}	Легкость инсталляции	3
F_{12}	Легкость эксплуатации	4
F_{13}	Разнообразные условия размещения	5
F_{14}	Простота изменений	5

Проверка расчетов с помощью FP-указателей. Проверка с помощью модели COSOMO II

- $FP = \text{Общее количество} * (0.65 + 0.01 * \sum_{i=1}^{14} F_i) = 318 * 1.17 = 372$

Значение производительности из метрического базиса:

$$\text{Производительность} = 2.55[FP/\text{чел.-мес.}]$$

Значения затрат и стоимости:

$$\text{Затраты} = FP/\text{Производительность}$$

$$\text{Стоимость} = \text{Затраты} * \$4500 = \$656500$$

- Проверка с помощью модели COSOMO II:

$$\text{Затраты} = A * \text{Размер}^B[\text{чел.-мес.}] = 2.5 * (33.3)^{1.16} = 145.87[\text{чел.-мес.}]$$

Номинальная длительность проекта

$$\text{Длительность} = [3.0 * (\text{Затраты})^{0.33+0.2(B-1.01)}] = 18[\text{мес.}]$$

Анализ чувствительности

Модель COCOMO II в задачах анализа чувствительности.

Пример: корпорация «СверхМобильныеСвязи» заказала разработку ПО для встроенной космической системы обработки сообщений.

- Ожидаемый размер ПО — 10 KLOC, используется серийный микропроцессор;
- Масштабные факторы имеют номинальные значения ($B = 1,16$) ;
- Автоматическая генерация кода не предусматривается;
- К разработке привлекаются главный аналитик и главный программист высокой квалификации - средняя оплата в команде = \$6000/месяц;
- У команды годовой опыт работы с предметной областью и полгода с нужной аппаратной платформой.

Анализ чувствительности. Таблица оценки пост-архитектурных факторов затрат для проекта

Фактор	Описание	Оценка	Множитель
RELY	Требуемая надежность ПО	Номинал.	1
DATA	Размер базы данных – 20 Кбайт	Низкая	0,93
CPLX	Сложность продукта	Очень высок.	1,3
RUSE	Требуемая повторная используемость	Номинал.	1
DOCU	Документирование жизненного цикла	Номинал.	1
TIME	Ограничения времени выполнения (70%)	Высокая	1,11
STOR	Ограничения оперативной памяти (45 Кбайт из 64 Кбайт, 70%)	Высокая	1,06
PVOL	Изменчивость платформы (каждые 6 мес.)	Номинал.	1
ACAP	Возможности аналитика (75%)	Высокая	0,83
PCAP	Возможности программиста (75%)	Высокая	0,87
AEXP	Опыт работы с приложением (1 год)	Номинал.	1
PEXP	Опыт работы с платформой (6 мес.)	Низкая	1,12
LTEX	Опыт работы с языком и утилитами (1 год)	Номинал.	1
PCON	Непрерывность персонала (12% в год)	Номинал.	1
TOOL	Активное использование программных утилит	Высокая	0,86
SITE	Мультиязычная разработка (телефоны)	Низкая	1,1
SCED	Требуемый график разработки	Номинал.	1
Множитель поправки M			1,088

Анализ чувствительности

Из таблицы видно, что увеличение затрат в 1,3 уравнивается их уменьшением вследствие высокой квалификации аналитика и программиста, а также активного использования программных утилит.

Расчет затрат и стоимости проекта:

$$\text{ЗАТРАТЫ} = A * \text{РАЗМЕР}^B * M_p = 2.5(10)^{1.16} * 1.088 = 39[\text{чел.-мес.}]$$

$$\text{СТОИМОСТЬ} = \text{ЗАТРАТЫ} * \$6000 = \$234000$$

Таковы стартовые условия программного проекта. Далее возможны несколько сценариев возможного развития событий

Сценарий понижения зарплаты

Рычаг — понижение квалификации аналитика и программиста. Тогда зарплата сотрудников снижается до \$5000. Оценки их возможностей становятся номинальными, а соответствующие множители затрат принимают единичные значения:

$$EM_{ACAP} = EM_{ACAP} = 1$$

В результате решения возрастает множитель поправки $M_p = 1,507$, а также затрат и стоимости:

$$\text{ЗАТРАТЫ} = 36 * 1,507 = 54[\text{чел.-мес.}]$$

$$\text{СТОИМОСТЬ} = \text{ЗАТРАТЫ} * \$5000 = \$270000$$

$$\text{Проигрыш_в_Стоимости} = \$36000$$

Спасибо за внимание