

Модели СОСОМО и СОСОМО II

Барышникова Марина Юрьевна
МГТУ им. Н.Э. Баумана
Каф. ИУ-7

baryshnikovam@mail.ru

Экономическая модель разработки ПО

Трудоемкость = (Персонал)(Среда)(Качество)(Размер^{Процесс})

Размер – размер конечного продукта (для компонентов, написанных вручную), который обычно измеряется числом строк исходного кода или количеством функциональных точек, необходимых для реализации данной функциональности. В это понятие также должны входить и другие создаваемые материалы, такие как документация, совокупность тестовых данных и обучающие материалы

Персонал – возможности персонала, участвующего в разработке ПО, в особенности его профессиональный опыт и знание предметной области проекта. Источниками сложностей могут быть требуемая надежность программного обеспечения, ограничения на производительность и хранение, требуемое повторное использование программных компонентов, а так же опыт работы программистов с данной средой программирования

Среда – состоит из инструментов и методов, используемых для эффективной разработки ПО и автоматизации процесса. Т.е. фактически, это приобретенная или потерянная эффективность вследствие уровня автоматизации процесса (большой уровень автоматизации приводит к уменьшению усилий и повышению эффективности)

Качество – требуемое качество продукта, что включает в себя его функциональные возможности, производительность, надежность и адаптируемость

Процесс – особенности процесса, используемого для получения конечного продукта, в частности, его способность избегать непроизводительных видов деятельности: переделок, бюрократических проволочек, затрат на взаимодействие



Модель оценки стоимости COSCOMO (COnstructive COst MOdel — конструктивная модель стоимости)

$$\text{Работа} = C1 * EAF * (\text{Размер})^{P1}$$

$$\text{Время} = C2 * (\text{Работа})^{P2}$$

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



Допущения модели COSOMO

- ▶ Исходные инструкции конечного продукта включают в себя все (кроме комментариев) строки кода, обрабатываемого компьютером
- ▶ Начало жизненного цикла проекта совпадает с началом разработки продукта, окончание — совпадает с окончанием приемочного тестирования, завершающего стадию интеграции и тестирования
- ▶ Работа и время, затрачиваемые на анализ требований, оцениваются отдельно, как дополнительный процент от разработки в целом
- ▶ Виды деятельности включают в себя только работы, направленные непосредственно на выполнение проекта
- ▶ Человеко-месяц состоит из 152 часов
- ▶ Проект управляется надлежащим образом, в нем используются стабильные требования



Режимы модели COSOMO

Название режима	Размер проекта	Описание	Среда разработки
Обычный	До 50 KLOC	Некрупный проект разрабатывается небольшой командой, для которой нехарактерны нововведения, разработчики знакомы с инструментами и языком программирования	Стабильная
Промежуточный	50 – 500 KLOC	Относительно небольшая команда занимается проектом среднего размера, в процессе разработки необходимы определенные инновации	Среда характеризуется незначительной нестабильностью
Встроенный	Более 500 KLOC	Большая команда разработчиков трудится над крупным проектом, необходим значительный объем инноваций	Среда состоит из множества нестабильных элементов



Формулы для оценки основных работ и сроков

Обычный вариант

$$\text{Работа} = 3,2 * \text{EAF} * (\text{Размер})^{1,05}$$

$$\text{Время (в месяцах)} = 2,5 * (\text{Работа})^{0,38}$$

Работа — количество человеко-месяцев

EAF — результат учета 15 уточняющих факторов (см. таблицу)

Промежуточный вариант

$$\text{Работа} = 3,0 * \text{EAF} * (\text{Размер})^{1,12}$$

$$\text{Время (в месяцах)} = 2,5 * (\text{Работа})^{0,35}$$

Размер — число исходных инструкций конечного продукта (измеряемое в тысячах строк кода KLOC)

Встроенный вариант

$$\text{Работа} = 2,8 * \text{EAF} * (\text{Размер})^{1,2}$$

$$\text{Время (в месяцах)} = 2,5 * (\text{Работа})^{0,32}$$



Значение драйверов затрат в модели СОСОМО

Идентификатор	Уточняющий фактор работ	Диапазон изменения параметра	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий
Атрибуты программного продукта							
RELY	Требуемая надежность	0,75-1,40	0,75	0,86	1,0	1,15	1,4
DATA	Размер базы данных	0,94-1,16		0,94	1,0	1,08	1,16
CPLX	Сложность продукта	0,70-1,65	0,7	0,85	1,0	1,15	1,3
Атрибуты компьютера							
TIME	Ограничение времени выполнения	1,00-1,66			1,0	1,11	1,50,
STOR	Ограничение объема основной памяти	1,00-1,56			1,0	1,06	1,21
VIRT	Изменчивость виртуальной машины	0,87-1,30		0,87	1,0	1,15	1,30
TURN	Время реакции компьютера	0,87-1,15		0,87	1,0	1,07	1,15
Атрибуты персонала							
ACAP	Способности аналитика	1,46-0,71	1,46	1,19	1,0	0,86	0,71
AEXP	Знание приложений	1,29-0,82	1,29	1,15,	1,0	0,91	0,82
PCAP	Способности программиста	1,42-0,70	1,42	1,17	1,00	0,86	0,7
VEXP	Знание виртуальной машины	1,21-0,90	1,21	1,1	1,0	0,9	
LEXP	Знание языка программирования	1,14-0,95	1,14	1,07	1,0	0,95	
Атрибуты проекта							
MODP	Использование современных методов	1,24-0,82	1,24	1,1	1,0	0,91	0,82
TOOL	Использование программных инструментов	1,24-0,83	1,24	1,1	1,0	0,91	0,82
SCED	Требуемые сроки разработки	1,23-1,10	1,23	1,08	1,0	1,04	1,1



Распределение работ и времени по стадиям жизненного цикла

<i>Вид деятельности</i>	<i>Работа (%)</i>	<i>Время (%)</i>
Планирование и определение требований	(+8)	(+36)
Проектирование продукта	18	36
Детальное проектирование	25	18
Кодирование и тестирование отдельных модулей	26	18
Интеграция и тестирование	31	28

Примечание: В основе распределения трудозатрат и времени лежит каскадная модель жизненного цикла



Декомпозиция работ по созданию ПО

<i>Вид деятельности</i>	<i>Бюджет (%)</i>
Анализ требований	4
Проектирование продукта	12
Программирование	44
Планирование тестирования	6
Верификация и аттестация	14
Канцелярия проекта	7
Управление конфигурацией и обеспечение качества	7
Создание руководств	6
Итого	100



Три различные модели оценки стоимости в 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 COSOMO II, которая используется после разработки архитектуры проекта. В состав этой модели включены новые драйверы затрат, новые правила подсчета строк кода, а также новые уравнения



Правила подсчета объектных точек

- ▶ количество изображений на дисплее. Простые изображения принимаются за 1 объектную точку, изображения умеренной сложности принимаются за 2 точки, очень сложные изображения принято считать за 3 точки
- ▶ количество представленных отчетов. Для простых отчетов назначаются 2 объектные точки, умеренно сложным отчетам назначаются 5 точек. Написание сложных отчетов оценивается в 8 точек
- ▶ количество модулей, которые написаны на языках третьего поколения и разработаны в дополнение к коду, написанному на языке программирования четвертого поколения. Каждый модуль на языке третьего поколения считается за 10 объектных точек

Модель ориентирована на применение объектных точек. Объектная точка — средство косвенного измерения ПО, для его расчета определяется количество экранов (как элементов пользовательского интерфейса), отчетов и компонентов, требуемых для построения приложения



Модель композиции приложения

$NOR = (\text{Объектные точки}) \times [(100 - \%RUSE) / 100]$ – новые объектные точки

$ЗАТРАТЫ = NOR / PROD$ [чел.-мес.]

PROD - оценка скорости разработки

Модель композиции приложения используется на этапе создания прототипов и анализа осуществимости

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 ранней разработки архитектуры

Работа = $2,45 * EArch * (Размер)^P$,

где

Работа — число человеко-месяцев

$EArch = PERS * RCPX * RUSE * PDIF * PREX * FCIL * SCED$

Размер — число функциональных точек (предпочтительно) или KSLOC

P — показатель степени

Время = $3,0 * (Работа)^{(0.33 + 0.2 * (P-1.01))}$

Примечание: Множитель EArch является произведением семи показателей, характеризующих проект и процесс создания ПО, а именно: надежность и уровень сложности разрабатываемой системы (RCPX), повторное использование компонентов (RUSE), сложность платформы разработки (PDIF), возможности персонала (PERS), опыт персонала (PREX), график работ (SCED) и средства поддержки (FCIL). Каждый множитель может быть оценен экспертно по шестибалльной шкале, либо его можно вычислить путем комбинирования значений более детализированных показателей, которые используются на постархитектурном уровне (см. след. слайд)



Модель ранней разработки архитектуры

Идентификатор	Составные драйверы затрат
Сложность продукта	RELY-DATA-CPLX-DOCU
Необходимость повторного использования	RUSE
Сложность платформы	TIME-STOR-PVOL
Опытность персонала	AEXP-PEXP-LTEX
Способности персонала	ACAP-PCAP-PCON
Возможности среды	TOOL-SITE
Сроки	SCED

Примечание: высокий опыт персонала и интенсивное повторное использование компонентов ведут к снижению затрат



Значения множителей трудоемкости, в зависимости от оценки их уровня

	Оценка уровня множителя трудоемкости					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверхвысокий
PERS	1.62	1.26	1.00	0.83	0.63	0.5
RCPX	0.60	0.83	1.00	1.33	1.91	2.72
RUSE	n/a	0.95	1.00	1.07	1.15	1.24
PDIF	n/a	0.87	1.00	1.29	1.81	2.61
PREX	1.33	1.22	1.00	0.87	0.74	0.62
FCIL	1.30	1.10	1.00	0.87	0.73	0.62
SCED	1.43	1.14	1.00	1.00	1.00	n/a



Пересчет ФР-оценок в ЛОС-оценки

Язык программирования	Количество операторов на один ФР
Ассемблер	320
С	128
Кобол	106
Фортран	106
Паскаль	90
С++ / Java / С#	53
Java Script	56
Ada 95	49
Visual Basic	32
Visual C++	34
Delphi Pascal	29
Perl	21
Prolog	54



Модель этапа постархитектуры

$$\text{Работа} = 2,45 * E_{\text{Арр}} * (\text{Размер})^p,$$

где:

Работа — число человеко-месяцев;

$E_{\text{Арр}}$ — результат применения семнадцати уточняющих факторов
постархитектурных этапов разработки

$$\text{Время} = 3,0 * (\text{Работа})^{(0.33 + 0.2 * (p - 1.01))}$$



Усовершенствованная постархитектурная модель COCOMO II

Идентификатор	Уточняющий фактор работ	Изменение в COCOMO II
RELY	Требуемая надежность	Без изменений относительно COCOMO
DATA	Размер базы данных	Без изменений относительно COCOMO
CPLX	Сложность продукта	Без изменений относительно COCOMO
RUSE	Требуемый уровень повторного использования	
DOCU	Документация	Добавлен. Определяет насколько документация соответствует требованиям жизненного цикла
TIME	Ограничение времени выполнения	Без изменений относительно COCOMO
STOR	Ограничение объема основной памяти	Без изменений относительно COCOMO



Усовершенствованная постархитектурная модель COCOMO II

Идентификатор	Уточняющий фактор работ	Изменение в COCOMO II
PVOL	Изменчивость платформы	Фактор изменчивости платформы
ACAP	Способности аналитика	Без изменений относительно COCOMO
AEXP	Знание приложений	Без изменений относительно COCOMO
PCAP	Способности программиста	Без изменений относительно COCOMO
PCON	Преемственность персонала	Новый параметр
LTEX	Знание языка программирования и инструментария	Изменен с целью охвата знаний инструментария и языка
SITE	Распределенная разработка. Взаимодействие между командами разработчиков	Новые параметры, определяющие степень взаимной удаленности команд разработчиков и степень автоматизации их деятельности
TOOL	Использование программных инструментов	Без изменений относительно COCOMO
SCED	Требуемые сроки разработки	Без изменений относительно COCOMO

Правила вычисления показателя степени в модели СОСОМО II

- ▶ Значение показателя степени изменяется от 1.1 до 1.24
- ▶ Значение показателя степени зависит от того, насколько новаторским является данный проект, от гибкости процесса разработки ПО, от применяемых процессов управления рисками, сплоченности команды программистов и уровня управления организацией-разработчиком
- ▶ Значение показателя степени рассчитывается с учетом пяти показателей по восьмибалльной шкале от низшего (7 баллов) до наивысшего (0 баллов) уровня
- ▶ Значения всех пяти показателей суммируются, сумма делится на 100, результат прибавляется к числу 1.01



Факторы, влияющие на показатель степени в модели СОСОМО II

- ▶ Наличие прецедентов у приложения: уровень опыта организации-разработчика в данной области
- ▶ Гибкость процесса: степень строгости контракта, порядок его выполнения, присущая контракту свобода внесения изменений, виды деятельности в течение жизненного цикла и взаимодействие между заинтересованными сторонами
- ▶ Разрешение рисков, присущих архитектуре: степень технической осуществимости, продемонстрированной до перехода к полномасштабному производству
- ▶ Сплоченность команды: степень сотрудничества и того, насколько все заинтересованные стороны (покупатели, разработчики, пользователи, ответственные за сопровождение и др.) разделяют общую концепцию
- ▶ Зрелость процесса: уровень зрелости организации-разработчика, определяемая в соответствии с моделью СММ



Новизна проекта (PREC)

Отражает предыдущий опыт организации в реализации проектов данного типа. Очень низкий уровень этого показателя означает отсутствие опыта, наивысший уровень указывает на компетентность организации-разработчика в данной области ПО

Характеристика	Рекомендуемое значение
Полное отсутствие прецедентов, полностью непредсказуемый проект	6,2
Почти полное отсутствие прецедентов, в значительной мере непредсказуемый проект	4,96
Наличие некоторого количества прецедентов	3,72
Общее знакомство с проектом	2,48
Значительное знакомство с проектом	1,24
Полное знакомство с проектом	0



Гибкость процесса разработки (FLEX)

Отображает возможность изменения процесса разработки ПО. Очень низкий уровень этого показателя означает, что процесс определен заказчиком заранее, наивысший — заказчик определил лишь общие задачи без указания конкретной технологии процесса разработки ПО

Характеристика	Рекомендуемое значение
Точный, строгий процесс разработки	5,07
Случайные послабления в процессе	4,05
Некоторые послабления в процессе	3,04
Большей частью согласованный процесс	2,03
Некоторое согласование процесса	1,01
Заказчик определил только общие цели	0



Разрешение рисков в архитектуре системы (RESL)

Отображает степень детализации анализа рисков, основанного на анализе архитектуры системы. Очень низкий уровень данного показателя соответствует поверхностному анализу рисков, наивысший уровень означает, что был проведен тщательный и полный анализ всевозможных рисков

Характеристика	Рекомендуемое значение
Малое (20 %)	7
Некоторое (40 %)	5,65
Частое (60 %)	4,24
В целом (75 %)	2,83
Почти полное (90 %)	1,41
Полное (100%)	0



Сплоченность команды (TEAM)

Отображает степень сплоченности команды и их способность работать совместно. Очень низкий уровень этого показателя означает, что взаимоотношения в команде сложные, а наивысший — что команда сплоченная и эффективная в работе, не имеет проблем во взаимоотношениях

Характеристика	Рекомендуемое значение
Сильно затрудненное взаимодействие	5,48
Несколько затрудненное взаимодействие	4,38
Некоторая согласованность	3,29
Повышенная согласованность	2,19
Высокая согласованность	1,1
Взаимодействие как в едином целом	0



Уровень зрелости процесса разработки (РМАТ)

Отображает уровень развития процесса создания ПО в организации-разработчике

Характеристика	Рекомендуемое значение
Уровень 1 CMM	7
Уровень 1+ CMM	6,24
Уровень 2 CMM	4,68
Уровень 3 CMM	1,12
Уровень 7 CMM	1,56
Уровень 5 CMM	0

