

*У статті описані результати дослідження мов високого рівня програмування, використовувані для розробки складних корпоративних інформаційних систем технологічної підготовки виробництва. Також наведені результати досліджень моделей оцінювання трудомісткості розробки програмних продуктів з точки зору їх застосування в корпоративних інформаційних системах технологічної підготовки виробництва*

*Ключові слова: життєвий цикл, клас, трудомісткість, КИС ТПП, програмний продукт, технічне завдання, прогнозування, вартість*

*В статье описаны результаты исследования языков высокого уровня программирования, используемые для разработки сложных корпоративных информационных систем технологической подготовки производства. Также приводятся результаты исследования моделей оценки трудоемкости разработки программных продуктов с точки зрения их применения в корпоративных информационных системах технологической подготовки производства*

*Ключевые слова: жизненный цикл, класс, трудоемкость, КИС ТПП, программный продукт, техническое задание, прогнозирование, стоимость*

# АНАЛИЗ МОДЕЛЕЙ РАСЧЕТА ТРУДОЕМКОСТИ ПРОГРАММНОГО ПРОДУКТА ПРИ РАЗРАБОТКЕ КИС ТПП

**И. Ш. Невлюдов**

Доктор технических наук, профессор, заведующий кафедрой\*

Контактный тел.: (057) 702-14-86

**В. В. Евсеев**

Кандидат технических наук, доцент, заместитель заведующего кафедры по научной работе\*

E-mail: evv\_crow@mail.ru

**С. С. Милютин**

Кандидат технических наук, доцент\*

E-mail: risyaya@rambler.ru

**В. О. Бортникова\***

E-mail: vika\_bortnikova@mail.ru

\*Кафедра технологии автоматизации РЭС и ЭВС

Харьковский национальный университет

радиоэлектроники

пр. Ленина, 14, г. Харьков, Украина, 61166

## 1. Введение

Для оценки трудоемкости разработки программного продукта для корпоративно - информационных систем технологической подготовки производства (КИС ТПП) существует много методов. Однако, сложность расчёта трудоемкости возникает на ранней стадии разработки технического задания для прогнозирования сроков и стоимости разрабатываемого программного продукта (ПП).

## 2. Анализ моделей оценки стоимости программного продукта

В связи с существующей актуальностью вопросом оценки стоимости программного продукта, разработаны математические модели, дающие количественную оценку и возможность создания новых моделей оценки стоимости программного продукта. Данные модели согласовываются с существующими жизненными циклами КИС ТПП, основные из которых представлены в стандарте ISO/IEC 12207:1995 «Information Technology – Software Life Cycle Processes» [3]. Наиболее известными и часто используемыми для оценки стоимости программного продукта являются модели:

- COCOMO I, (Constructive Cost Model) – basis model;

- COCOMO II – intermediate model;

- COCOMO III – Advanced/Detailed model;

- SLIM model (Модель Патнэма).

Проведем анализ возможности и эффективности использования основных моделей для оценки КИСС ТП на ранних этапах разработки. Важным критерием является подход, использованный в данных моделях, а также используемые математические модели и выведенные формулы описания процесса создания ПП на математическом языке. Модели методов оценки реализованы в программных продуктах различных фирм (QSM, Costar Software и т.д.).

COCOMO (Constructive Cost Model – basismodel) – конструктивная модель стоимости, основанная на том, что величина трудоемкости разработки программных систем (человек/месяц) зависит от многих факторов. Наибольшее влияние на величину трудоемкости оказывает объем программного продукта (число исходных команд), который изменяется в широком диапазоне, зависит от выбора языка и среды программирования, а также длина программного кода – не всегда показывающая качество программы.

В общем виде уравнение моделей имеет вид:

$$E = a \times S^b \times EAF, \quad (1)$$

где E – расходы на проект (человек-месяц);

S – размер кода (KLOC);

EAF – фактор уточнения расходов (Effort Adjustment Factor);

a, b – параметры, зависящие от вида дополнения.

В модели COSOMO используется ряд допущений:

1) Исходный код конечного продукта включает в себя все (кроме комментариев) строки кода.

2) Начало цикла разработки 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 оценки разработки в целом).

3) Виды деятельности включают в себя только непосредственно направленные на выполнение проекта работы, в них не входят обычные вспомогательные виды деятельности, такие, как административная поддержка, техническое обеспечение и капитальное оборудование.

4) Человеко-месяц состоит из 152 ч.

5) Проект управляется надлежащим образом, в нем используются стабильные требования.

Наиболее фундаментальные расчеты в модели COSOMO является использование уравнения для оценки человек-месяцев, необходимых для разработки проекта. При оценке непосредственных затрат и длительности полного цикла разработки сложных программных продуктов, объем программ используется в качестве базового (главного) параметра, остальные же факторы можно отражать поправочными коэффициентами. В модели COSOMO введено 15 поправочных факторов, относящихся к одной из четырех категорий, которые в свою очередь получают оценку по 6-балльной шкале:

- атрибуты продукта (его сложность и требования к его надежности, размер БД, сложность архитектуры приложения);

- атрибуты системы (ограничение на оперативную память и время выполнения, время компиляции, надежность используемых виртуальных машин);

- атрибут команды разработчиков (знание прикладной области, аналитические способности, опыт разработки, опыт в данном языке программирования);

- атрибуты проекта (используемые средства разработки, применение методов разработки, применение методов разработки программного обеспечения, системы контроля разработки приложения).

В COSOMO модели предложено три уровня модели [1]:

1. Базовая COSOMO (basis model) модель является стационарной моделью, которая вычисляет затраты, необходимые для разработки программного обеспечения как функцию размера программы, выраженную в оценочном множестве доставленных исходных кодов инструкций (KDSI).

2. Промежуточная модель COSOMO (intermediate model) вычисляет затраты, необходимые для разработки программного обеспечения, выраженные в виде функции размера программы и набора из 15 поправочных коэффициентов, которые включают субъективные оценки продукции, оборудования, персонала, а также проектные атрибуты.

3. Расширенная или подробная COSOMO (Advanced/Detailed model) модель включает в себя все характеристики промежуточных версий с оценкой воздействия затрат на каждом шаге (анализ, проектирование и т.д.) в процессе разработки.

COSOMO моделей зависит от двух основных уравнений:

- усилий в области развития (MM - man-month):

$$MM = a \times KDSI^b, \tag{2}$$

где MM – человеко-месяц,

KDSI – функция размера программы, выраженная в оценочном множестве доставленных исходных кодов инструкций;

b – поправочный коэффициент, зависящий от уровня COSOMO модели.

Усилия и время на разработку (effort and development time (TDEV)), определяются по формуле:

$$TDEV = 2.5 \times MM^c, \tag{3}$$

где TDEV – усилия и время на разработку;

MM – человеко-месяц;

c – коэффициент, зависящий от выбора модели.

Коэффициенты a, b и c зависят от модели разработки. Есть три модели разработки (табл. 1).

**Таблица 1**

**Модели разработки и характеристики проекта**

Модель разработки			Характеристики проекта	
Тип модели	Размер	Инновация	Срок\ограничение	Среда разработки
Organic (основной)	маленький	небольшая	не компактное	стабильная
Semi-detached (парный)	средний	средняя	среднее	средняя
Embedded (встроенный)	большой	большая	компактное	аппаратный комплекс\интерфейс пользователя

К базовой COSOMO модели относится параметризация уравнения без особого детального рассмотрения проекта.

В базовой модели COSOMO используются те же основные уравнения. Пятнадцать поправочных коэффициентов оцениваются по шкале от «очень низких» до «очень высоких» и рассчитываются как множители конкретного усилия, каждый из них возвращает поправочный коэффициент умножения, который дает в общем фактор уточнения расходов EAF (Effort Adjustment Factor). Для стоимости затрат поправочный коэффициент равен 1 и оценивается, как нормальный.

В расширенной COSOMO модели (ADVANCED, DETAILED COSOMO) вычисляются усилия в зависимости от размера программы и набор поправочных

коэффициентов, которые взвешены в соответствии с каждой фазой жизненного цикла программного обеспечения. Расширенная модель применяется в промежуточной модели на уровне компонентов.

Таблица 2

Параметры базовой COSOMO

Базовая COSOMO	a	b	c
Organic (основной)	2,4	1,05	0,38
Semi-detached (парный)	3,0	1,12	0,35
Embedded (встроенный)	3,6	1,20	0,32

В промежуточной модели COSOMO (INTERMEDIATECOSOMO) используются те же основные уравнения, но пятнадцать драйверов затрат оцениваются по шкале от «очень низких» до «очень высоких» для расчета мультипликатора усилий и каждый из них возвращает поправочный коэффициент умножения, который дает в общем EAF.

Поправочный коэффициент равен 1 для стоимости затрат. В дополнение к EAF, параметр модели «a» немного отличается в промежуточной COSOMO от базисной модели. Параметр «b» остается тем же в обеих моделях.

Таблица 3

Параметры промежуточной COSOMO

Средняя COSOMO	A	b	c
Organic (основной)	3,2	1,05	0,38
Semi-detached (парный)	3,0	1,12	0,35
Embedded (встроенный)	2,8	1,20	0,32

Основное уравнение промежуточной COSOMO является коррекцией человеко-месяца:

$$MM_{\text{корт}} = EAF \times MM_{\text{nominal}} \quad (4)$$

В расширенной или детальной COSOMO модели (ADVANCED, DETAILED COSOMO) вычисляются усилия, функции программы, размер и набор затрат драйверов которых взвешиваются в соответствии с каждой фазой программного обеспечения ЖЦ. Расширенная модель относится к средней модели на уровне компонентов, только этап подхода используется для объединенной оценки.

Четыре фазы, используемые в расширенной модели COSOMO:

- требования планирования и проектирования изделий (RPD – planning and product design);
- детальное проектирование (DD – detailed design);
- код и модульного тестирования (CUT – code and unittest);
- интеграцию и тестирование (IT – integration and test).

Каждый затратный драйвер разбивается по фазам, как в примере, приведенном в таблице 4. Оценки для каждого модуля объединяются в подсистемы и в конечном итоге получается общая оценка проекта.

Использование подробной затраты драйверов, оценка определяются для каждого этапа ЖЦ.

Модель Патнэма (SLIM model) создана для проектов объемом больше 70 000 строк кода. Используя статистический анализ проектов, Патнэм обнаружил, что взаимосвязь между тремя основными параметрами проекта (размером, временем и трудоемкостью) напоминает функцию Нордена-Рэля, отражающую распределение трудовых ресурсов проекта в зависимости от времени. Модель основана на том, что затраты на разработку ПО распределяются согласно кривым Нордена-Рэйли, которые являются графиками функций.

Таблица 4

Аналитик возможности (ACAP – Analyst capability) усилия, множитель для расширенной COSOMO

Затратный драйвер	Рейтинг	RPD	DD	CUT	IT
ACAP	Очень низкий	1,80	1,35	1,35	1,50
	Низкий	0,85	0,85	0,85	1,20
	Нормально	1,00	1,00	1,00	1,00
	Высокий	0,75	0,90	0,90	0,85
	Очень высокий	0,55	0,75	0,75	0,70

Функция Нордена-Рэля моделируется дифференциальным уравнением:

$$\frac{dy}{dt} = 2 \times K \times a \times t \times e^{-at^2}, \quad (5)$$

где  $\frac{dy}{dt}$  - скорость роста персонала проекта;  
t - время, прошедшее от начала проекта до изъятия продукта из эксплуатации;

K - область под кривой, представляет полную трудоемкость в течение всего жизненного цикла (включая сопровождение), выраженную в человеко-годах;  
a - константа, которая определяет форму кривой (фактор ускорения) и вычисляется по формуле:

$$a = \frac{1}{2 \times t_d^2}, \quad (6)$$

где  $t_d$  – время разработки.

Приняв ряд допущений, Патнэм получил следующее уравнение [2]:

$$E = 0.4 \times [s/c]^3 \times \frac{1}{t_d^4}, \quad (7)$$

где E - трудоемкость разработки ПО;  
S - размер ПО в LOC коде;  
 $t_d$  - планируемый срок разработки;  
C - технологический фактор, учитывающий различные аппаратные ограничения, опыт персонала и характеристики среды программирования. Этот фактор определяется на основе хронологических данных по прошлым проектам и, согласно рекомендациям Патнэма, определяется для различных типов проектов следующим образом:

- проект, внедренный в сжатые сроки без детальной проработки, - 1500;

- проект, выполненный в соответствии с четким планом, - 5000;  
 - проект, предусматривающий оптимальную организацию и поддержку, - 10000.

Оптимальный срок разработки определяется как:

$$t_d = 2.4 \times E^{1/3}, \quad (8)$$

где E - трудоемкость разработки ПО;

- не учитывают сложность языка программирования, уровень интеграции в КИС ТПП, что не дает возможности оценить время написания исполняющего кода программного продукта в зависимости от сложности функций и процедур;

- не учитывают возможность применения современных промышленных систем управления базами данных, что не позволяет оценить сложность запросов на языке SQL, количество процедур, триггеров и степень их наследования;

- позволяют рассчитывать трудоемкость на последнем этапе разработки программного продукта, таким образом увеличивая риск создания «мёртвого проекта» или существенно увеличивая его стоимость относительно изначальной.

---

### 3. Выводы

---

В ходе анализа существующих моделей расчета трудоемкости можно сделать вывод о том, что предложенные модели:

---

### Литература

1. Евсеев В.В. Применение программных метрик кода на раннем этапе жизненного цикла программного обеспечения [Текст] / В.В. Евсеев // Восточно-европейский журнал передовых технологий. – Харьков, 2011. – № 1/2 (49). – С.19–21.
2. Фатрелл, Р. Управление программными проектами: достижение оптимального качества при минимуме затрат [Текст] / Р. Фатрелл, Д. Шафер, Л. Шафер; пер. с англ. – М.: Вильямс, 2003.
3. Невлюдов, И.Ш. Модели жизненного цикла программного обеспечения при разработке корпоративных информационных систем технологической подготовки производства [Текст] / И.Ш. Невлюдов, В.В. Евсеев, В.О. Бортникова // Вісник Національного технічного університету «ХПІ». Збірник наукових праць. Тематичний випуск: Нові рішення в сучасних технологіях. – Харків: НТУ «ХПІ», 2011. – №2. – С.94–101.

### Abstract

*The article describes the results of study of assessment models of software products development from the point of view of their application in corporate information systems of engineering process. The main models COCOMO I, II, III and SLIM model were examined. The conclusion is that these models can be applied only to assess the laboriousness of software products on the final stage of the life cycle, which prevents the assessment of their cost during the designing of requirements specification. It should be noted that it increases the risk of “dead software product”. Another risk is the failure to implement all the functions and changes of the code, as well as works with databases. The need for such changes may be caused by the amendment of the interface and functional of a software product, which can be added by a client after the approval of requirements specification*

**Keywords:** *life cycle, class, laboriousness, corporate information systems of engineering process, software product, requirements specification, forecasting, cost*