

# Моделювання в економіці, організація виробництва та управління проектами

УДК 338.512 : 004.4

М.А. Голованова, Е.В. Надин

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков*

## ОЦЕНКА ТРУДОЕМКОСТИ РАБОТ НА РАННИХ СТАДИЯХ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Публикация посвящена вопросам оценки трудоемкости работ на ранних стадиях создания программного обеспечения с учетом множества факторов в условиях ограниченности исходной информации. Рассмотрены методы определения трудозатрат на основе современных ППП, особое внимание уделено методам на основе учета количества строк кода. Предложен подход, учитывающий этапы жизненного цикла создания ПО в разрезе декомпозиции основных категорий и этапов работ, а также влияющие факторы.*

**Ключевые слова:** программное обеспечение, трудозатраты, объем работ, методы оценки стоимости.

### Введение

Разрабатывая новый программный продукт, необходимо правильно рассчитать проектную стоимость, которая напрямую зависит от сроков его исполнения. Развитие моделей, методов и средств оценки стоимости программного обеспечения (ПО) сегодня достигло уровня практического применения. Однако они не используются при разработке ПО из-за отсутствия информации, средств и специалистов в Украине. Существуют трудности в оценке затрат на разработку программного обеспечения, так как необходимо учитывать множество разнообразных факторов: сложность и объем решаемой задачи, стоимость и доступность необходимых ресурсов, необходимый уровень абстракции при разработке и др.

Недооценка стоимости, времени и других ресурсов, требуемых для создания программного проекта, влечет за собой недостаточную численность проектной команды, чрезмерно сжатые сроки разработки и, как результат, утрату доверия к разработчикам в случае нарушения графика. С другой стороны, перестраховка и переоценка, т.е. если для проекта выделено больше ресурсов, чем реально необходимо, причем без должного контроля за их использованием, то такой проект окажется более дорогостоящим, чем должен был быть при грамотной оценке, и приведет к запаздыванию с началом следующего проекта.

### 1. Особенности учета трудозатрат в существующих методах и моделях оценки стоимости ПО

Методы и модели оценки стоимости ПО можно разделить на две группы: неалгоритмические методы, в которых при оценке стоимости ПО использу-

ются определенные схемы и принципы, а не математические формулы, и алгоритмические модели, представляющие собой одну или несколько функций, описывающие зависимость между характеристиками проекта и затратами на его реализацию. К неалгоритмическим методам относятся Price-to-win, оценка по Паркинсону, экспертная оценка, оценка по аналогии. К алгоритмическим моделям относятся модель Путнэма SLIM (степенная, аналитическая) и модель СОСОМО (степенная, эмпирическая) [1–3].

Такие методы оценки применимы, если требования будущей системы документированы и существует возможность определить функциональные характеристики разрабатываемого ПО. Однако во многих случаях оценка затрат проводится только на основе предварительных требований к ПО при наличии минимума информации. Как правило, независимые оценки стоимости ПО (т.е. выполненные людьми, которые никак не зависят от команды разработчиков) обычно неточны, но при совместной работе менеджера проекта и ведущих специалистов по созданию архитектуры, разработке и тестированию, достоверность оценок трудоемкости и стоимости ПО повышается. Большое значение имеют критерии и метрики, позволяющие оценивать затраты на создаваемый продукт, в том числе и трудовые затраты.

Рассмотрим более подробно вопросы трудоемкости создания ПО как одного из основных влияющих факторов на его стоимость. Оценка трудоемкости производится на основе размера по такой процедуре:

- оценка размера разрабатываемого ПО;
- оценка трудоемкости в человеко-часах;
- оценка продолжительности проекта по созданию ПО в календарных месяцах;
- оценка стоимости проекта.

Оценка размера программного проекта базируется на знании требований к системе. Для такой оценки существуют два основных способа:

1) по аналогии с подобными проектами и его известными оценками;

2) путем подсчета размера по определенным алгоритмам на основе исходных данных – требований к системе.

Основными единицами измерения размера программного проекта являются:

- количество строк кода (LOC–LinesofCode);
- функциональные точки (FP–FunctionPoints).

Количество строк кода – самая известная и до недавнего времени распространенная единица измерения. Однако при ее использовании возникает ряд вопросов и проблем [4]:

1. Каким образом можно определить количество строк кода до того, как они фактически будут написаны либо просто спроектированы?

2. Как показатель количества строк кода может отражать величину трудозатрат, если не будут учитываться сложность проекта, способности (стиль) программиста либо возможности применяемого языка программирования?

3. Каким образом разница в количестве строк кода может быть трансформирована в объем эквивалентной работы?

Несмотря на то, что строки кода как единицы измерения являются наиболее широко используемыми, указанные вопросы не способствуют дальнейшей популяризации строк кода в качестве метрик. При этом взаимосвязь между LOC и затрачиваемыми усилиями не является линейной. Средняя производительность работы программистов за двадцать лет осталась неизменной (в том числе и на новых языках) и составляет около 3000 строк кода на одного программиста в год [5], т.е. уменьшение времени, затрачиваемого на цикл разработки, не может быть достигнуто за счет значительного повышения производительности труда программистов. Причем это не зависит от усовершенствований языка программирования, усилий со стороны менеджеров или сверхурочных работ. На самом деле первостепенное значение имеет набор функциональных свойств и качество программного продукта, а не количество строк кода.

В последнее время широкое развитие приобрели методика оценки трудоемкости на основе функциональных точек и методика оценки трудоемкости на основе вариантов использования. Но и эти подходы имеют свои достоинства и недостатки.

## 2. Трудоемкость работ на этапах создания ПО

Увязывая стоимость ПО с этапами жизненного цикла создания ПО, можно разработать свои подходы к оценке трудоемкости на каждом этапе, с последую-

щим суммирование всех полученных оценок. Предлагается выделить для оценки стоимости разработки и внедрения ПО такие последовательные этапы:

1) анализ функциональных и нефункциональных требований, требований к внешним интерфейсам ПО, к аппаратному обеспечению, по обеспечению информационной безопасности, определение архитектуры аппаратного обеспечения, проектирование ПО, идентификация объектов и их иерархия;

2) определение структурных единиц и стоимости поставок (аренды) программного и аппаратного обеспечения;

3) определение размера программной части ПО и составление документации по оценке размера программной части;

4) оценка объема работ – идентификация рисков, разбиения работ, расчет и корректировка объема работ и трудозатрат по разработке ПО;

5) расчет объема работ и трудозатрат на кодирование и тестирование, расчет длительности кодирования и тестирования, составление плана работ по разработке ПО и сценариев тестирования;

6) определение количественных, качественных, внутренних и внешних характеристик качества и оценка качества ПО;

7) оценка общей стоимости проекта;

8) оценка влияния рисков, ее регулирование, а также регулирование затрат, основанных на оценке рисков;

9) подтверждение и корректировка оценки стоимости рисков;

10) разработка плана и задач внедрения, реализация полномасштабного внедрения ПО.

Как было отмечено выше, для оценки затрат труда на создание ПО, продолжительности самих работ и стоимости, необходимо основываться на статистике из прошлого опыта. В случае отсутствия таких данных, или, если ведется работа над новым проектом, необходимо применение специальных методик, позволяющих предварительно оценить трудозатраты на планируемый программный проект.

Среди факторов, влияющих на размер и объем работ по созданию ПО, можно выделить: новые технологии: код, язык программирования или метод проектирования, низкий уровень готовности технологий – истечение срока действия лицензий, моральное устаревание технологий, появление принципиально новых технологий, которые сразу делают определенные продукты и услуги неактуальными; оптимистические предположения, связанные с наследованием элементов ПО; возможность многократного использования кода – увеличение количества ошибок в связи с неправильно составленным многократно используемым программным кодом; и ряд других.

Поскольку разработка ПО охватывает программных инженеров, системотехников, тестирующих

ков, программістів, а також включає в себе розробку та аналіз вимог по інтегруванню та тестуванню ПО, то особе увагу при оцінці трудозатрат слід уділяти саме цим категоріям.

Етап аналізу та проектування являються визначальними при створенні ПО. Можливо виділити три фази цього аналізу:

- 1) дослідження та системний аналіз існуючого ПО та виявлення недоліків;
- 2) узагальнення результатів системного аналізу та створення попередньої концепції нового або модернізованого ПО;
- 3) розробка системного проекту комплексу програм та баз даних, визначаючих методи та засоби подальшого детального проектування та всього життєвого циклу ПО та бази даних.

Трудозатрати по створенню ПО повинні включати витрати праці фахівців сторонніх організацій, в відповідності з умовами укладеного контракту, а також власних аналітичних структур. Обсяг робіт по створенню ПО може бути визначений згідно контракту, укладеному зі спеціалістами сторонніх організацій, в якому вказується обсяг виконуваних робіт та строки їх виконання, а також згідно посадовими інструкціями власного персоналу.

### 3. Оцінка трудоемкости работ основных этапов жизненного цикла создания ПО

Объем работ по созданию ПО в разрезе декомпозиции основных категорий и этапов можно определить в процентном соотношении с использованием статистического анализа SEER-SEM [6] (табл. 1) или согласно какому-либо другому статистическому методу.

Таблица 1  
Декомпозиция разработки ПО

| Категория работ                              | % от объема работ по разработке ПО |
|--|------------------------------------|
| Разработка ПО:                               | 100%                               |
| Анализ требований ПО                         | 12%                                |
| Разработка ПО                                | 55%                                |
| Внедрение ПО                                 | 13%                                |
| Программное интегрирование и тестирование ПО | 20%                                |

Преобразование размера каждой программной функции в объем работ по разработке ПО можно вычислить следующим образом:

- 1) расчет объема работ по разработке ПО предлагается выполнять по формуле (1) и определять в рабочих месяцах

$$Q_{ПО} = R / w, \quad (1)$$

где  $w$  – программная производительность, определяется в SLOC/PM;  $R$  – полученный размер, определяется в SLOC.

Для определения программной производительности используются данные из аналогичного проекта по созданию ПО. Если данные аналогичного проекта недоступны, можно использовать табл. 2. Показатели производительности, представленные в табл. 2, отражают процесс разработки ПО.

Таблица 2  
Производительность программной разработки для средних промышленных проектов

| Характеристика          | Производительность программной разработки (SLOC/PM) |
|-------------------------|---|
| Классические показатели | 130–195   |
| Развивающиеся методы    | 244–325   |
| Новое встроенное ПО     | 17–105  |

- 2) корректировку объема работ по оценке каждой категории работ по созданию ПО предлагается выполнять по формуле (2), и определять в рабочих месяцах

$$Q_{ПО}^{корр} = Q_{ПО} k_Q, \quad (2)$$

где  $k_Q$  – множитель объема работ, определяется по данным табл. 3.

Таблица 3  
Множители объема работ для программного наследования

| Категория программного наследования                            | Множитель объема работ |
|--|------------------------|
| Новый проект и новый код                                       | 1,2                    |
| Аналогичный проект и новый код                                 | 1,0                    |
| Аналогичный проект и код, используемый многократно             | 0,8                    |
| Аналогичный проект и расширенный код, используемый многократно | 0,6                    |

- 3) скорректированный объем работ каждой функциональной и программной категории наследования необходимо просуммировать, чтобы определить общий объем работ.

Трудозатраты на разработку ПО могут включать оплату работ специалистов сторонних организаций, в соответствии с условиями заключенного контракта, а также оплату работ собственного персонала. Так следует учитывать затраты на аутсорсинг по разработке программного обеспечения, который может включать следующие виды услуг:

- 1) разработка ПО;
- 2) поддержка ПО;
- 3) рефакторинг и реинжиниринг приложений;
- 4) внедрение разрабатываемого ПО;
- 5) документирование ПО;
- 6) обучение персонала заказчика;
- 7) консалтинг в области ИТ.

При принятии решения об использовании аутсорсинга по разработке ПО необходимо учитывать следующие факторы: стоимость собственных разра-

боток в соотношении со стоимостью услуг аутсорсинга; риски нарушения режима информационной безопасности; риски разрыва отношений с аутсорсинговой компанией и дальнейшее развитие созданного приложения и др. Аутсорсинг позволяет использовать опыт профессионалов, сократить риски и снизить затраты, увеличивая отдачу вложенных средств.

Оценка трудозатрат на разработку ПО *собственным персоналом* определяется по количеству

строк кода. Базовая формула оценки трудозатрат с использованием модели COCOMO II:

$$E = a \cdot KSLOC^b \cdot EAF, \quad (3)$$

где  $E$  – трудозатраты, выраженные в рабочих месяцах;  $a$ ,  $b$  – коэффициенты, которые зависят от режима использования модели COCOMO II (табл. 4);  $KSLOC$  – количество тысяч строк кода;  $EAF$  – фактор корректировки трудозатрат.

Таблица 4

Режимы использования и базовые значения коэффициентов модели COCOMO II

| Название режима | Размер проекта  | Описание  | Значение коэффициента $a$ | Значение коэффициента $b$ |
|-----------------|-----------------|---|---------------------------|---------------------------|
| Органичный      | До 50 KSLOC     | Некрупный проект разрабатывается небольшой командой, для которой не характерны нововведения и среда остается стабильной   | 2,4                       | 1,05                      |
| Сблокированный  | 50–300 KSLOC    | Относительно небольшая команда занимается проектом среднего размера, в процессе разработки необходимы определенные инновации, среда характеризуется незначительной нестабильностью  | 3,0                       | 1,12                      |
| Внедренный      | Более 300 KSLOC | Большая команда разработчиков трудится над крупным проектом, необходим значительный объем инноваций, среда состоит из множества элементов, которые не характеризуются стабильностью | 3,6                       | 1,20                      |

Фактор корректировки трудозатрат  $EAF$  увеличивает или уменьшает трудозатраты в зависимости от факторов среды разработки. Расчет фактора корректировки трудозатрат предлагается выполнять как

$$EAF = \prod_{i=0}^n C_i, \quad (4)$$

где  $C_i$  – один из факторов среды разработки.

Среди факторов корректировки трудозатрат  $EAF$  следует выделить:

1) фактор учета технологии разработки, который учитывает увеличение количества трудозатрат в результате вовлечения разного количества сотрудников, рабочих месяцев:

$$F_{\text{техн}} = \frac{A(1 + K_{\text{test}}) + C \cdot K_{\text{test}}}{B} + 1 + K_{\text{test}}, \quad (5)$$

где  $A$ ,  $B$ ,  $C$  – количество сотрудников категорий  $A$ ,  $B$ ,  $C$  (**категория «А»:** менеджер проекта; менеджер разработки программной системы; менеджер внедрения ПО; менеджер эксплуатации; менеджер сопровождения ПО; аналитик бизнес-процессов; бизнес-проектировщик; системный аналитик; технолог бизнес-процессов; **категория «В»:** программист; проектировщик программной системы; проектировщик пользовательского интерфейса; интегратор; технический писатель; системный администратор; администратор баз данных; специалист управления конфигурацией; **категория «С»:** тестировщик; тестировщик службы сопровождения; разработчик тестов; аналитик службы сопровождения).

Расчет длительности разработки выполняется по формуле (6), выражается в рабочих месяцах:

$$L = 1(1 + K_{\text{test}}). \quad (6)$$

Время тестирования является величиной зависимой от времени кодирования и определяется коэффициентом  $K_{\text{test}}$ :

$$\frac{l_{\text{test}}}{l} = K_{\text{test}}, \quad (7)$$

где  $l$  – продолжительность процесса кодирования, выраженная в рабочих месяцах;

$l_{\text{test}}$  – продолжительность процесса тестирования, выраженная в рабочих месяцах.

2) фактор учета сложности и опыта разработки  $F_{\text{сложн}}$  определяется эмпирически и состоит из значений, приведенных в табл. 5.

Этот способ оценки используется на начальной стадии программного проекта для оценки всего проекта в целом. Для небольших проектов и небольших по объему работ его использование затруднительно.

На стадии «Завершение оценки объема работ» следует провести охват всех рассмотренных ранее категорий работ, а также учесть дополнительные объемы работ по разработке ПО (табл. 6).

#### 4. Определение трудозатрат на кодирование и тестирование

При определении периода времени и расчета трудозатрат на кодирование и тестирование, которые необходимы для завершения программного проекта, и сроков выполнения рабочих элементов работ за основу оценки трудозатрат берется оценка трудозатрат по количеству строчек кода по формулам (4) и

(5). Оценка каждого функционального требования выполняется отдельно, при этом значение константы  $b$  принимается равной 1, так как функциональное требование является составной частью проекта. Объем работ зависит от вида работы, а время его выполнения зависит от программной производительности. Формулу (4) следует преобразовать к виду:

$$E = \sum_i \sum_j \left( (KSLOC_{ij} / W_j) \cdot EAF \right), \quad (8)$$

где  $W_j$  – программная производительность, необходимая для выполнения  $j$ -го вида работы;  $KSLOC_{ij}$  – количество тысяч строк кода  $j$ -го вида работ для реализации  $i$ -го требования.

Таблица 5  
Базовые значения коэффициентов фактора учета сложности и опыта разработки

| Фактор учета сложности и опыта разработки | Значение |
|---|----------|
| Надо поправить то, что есть               | 1,00     |
| Уже делали раньше, есть опыт              | 1,10     |
| Есть опыт и не видно трудностей           | 1,30     |
| Опыта нет, но есть помощь                 | 2,00     |
| Нет ни опыта, ни помощи                   | 4,00     |

Таблица 6  
Дополнительный объем работ по разработке ПО

| Категория работ  | Дополнительный объем работ по разработке ПО, %                  |
|--|---|
| Управление ПО  | Прибавить 6–27%   |
| Системный уровень поддержки тестирования                     | Прибавить 34–112%   |
| Система обеспечения качества                                 | Прибавить 6–11%   |
| Независимая проверка   | Прибавить 9–45%   |
| Дополнительная деятельность:                                 |   |
| управление конфигурацией проектов                            | Прибавить 3–6%  |
| управление проектами   | Прибавить 8–11%   |
| управление поставками программного и аппаратного обеспечения | Прибавить 11–22%  |
| доработка  | Прибавить 17–22%  |
| Эксплуатация, первые пять лет                                | Прибавить 22% объёма работ по разработке ПО за год эксплуатации |

Производительность  $W$  определяется опытным путем, исходя из опыта работы над предыдущими компонентами.

Длительность кодирования и тестирования, выраженную в рабочих месяцах, можно вычислить так:

$$T = \sum_i \sum_j \left( \frac{KSLOC_{ij}}{W_i} F_{сложн_i} \right) \frac{1}{P} (1 + K_{test}).$$

Трудозатраты для определения стоимостной оценки кодирования и тестирования, выраженные в рабочих месяцах, вычисляются по формуле

$$E = \sum_i \sum_j \left( (KSLOC_{ij} / W_j) \cdot F_{сложн_i} \right) F_{техн}.$$

При определении трудозатрат на тестирование важно:

- учитывать трудозатраты персонала категории «С» на составление планов;
- распределить время для каждой категории работ;
- определить допустимую рабочую нагрузку;
- после составления плана работ проверить распределение работ в соответствии с аналогичным опытом, используя табл. 7 и 8. Значимые отклонения от этих данных приведут к увеличению риска и повышению стоимости.

При апробации методики следует учитывать расчеты комплексных оценок индивидуального труда разработчиков программного обеспечения.

Практическая значимость представленной методики заключается в том, что предложенный алгоритм оценки трудоемкости работ по созданию ПО на ранних стадиях разработки может быть использован при разработке регламентов управленческих решений с учетом стратегических и тактических задач организации.

Таблица 7  
Распределение времени по фазам разработки ПО

| Фаза                                      | Промышленные данные (средства) |
|---|--------------------------------|
| Анализ требований                         | 18%                            |
| Разработка ПО                             | 22%                            |
| Внедрение                                 | 36%                            |
| Программное интегрирование и тестирование | 24%                            |

Таблица 8  
Распределение объема работ для нового, модифицированного или преобразованного ПО

| Фаза                         | Новое ПО | Существующее модифицир. ПО | Преобразованное ПО |
|------------------------------|----------|----------------------------|--------------------|
| Анализ требований проекта    | 20%      | 15%                        | 5%                 |
| Рабочий проект, код и тест   | 57%      | 10%                        | 5%                 |
| Интеграция и тестирование ПО | 23%      | 40%                        | 30%                |
| Относительный объем работ    | 100%     | 65%                        | 40%                |

## Выводы

Таким образом, в статье предложен подход определения трудозатрат на создание программного обеспечения с учетом разбивки работ по этапам жизненного цикла создания ПО. При расчете затрат учтены влияющие факторы, которые направлены на уточнение (увеличение или уменьшение) величины трудозатрат, а также учтена интенсивность участия различных категорий работников как собственного, так и стороннего персонала. Предложенный подход позволит оценивать трудоемкость работ по созданию ПО на ранних стадиях проектирования.

**Список літератури**

1. Parkinson S.N. *Parkinson's Law and Other Studies in Administration [Текст] / S.N. Parkinson.* – Houghton-Mifflin, 1957. – 148 p..
2. Boehm B.W. *The COCOMO 2.0 Software Cost Estimation Model [Текст] / B.W. Boehm.* – American Programmer, 2000. – 586 p.
3. Heires J. *More with Less: SLIM-Estimate 5.0 Product Review [Текст] / J. Heires, T. Doing.* – QSM Software – 2002. – 46 p.
4. Сидоров Н.А. *Модели, методы и средства оценки стоимости программного обеспечения / Н.А. Сидоров,*

Д.В. Баценко, Ю.Н. Василенко, Ю.В. Шебетин // Пробл. програмув. – 2006. – 2-3 [спец. вип.]. – С. 290-298.

5. David L. *Norden-Raleigh Analysis: A Useful Tool for EVM in Development projects [Текст] / L. David.* – The Measurable News. – 2002. – 24 p.
6. SEER-SEM [Электронный ресурс]. – Электрон.дан. – Режим доступа к ресурсу: <http://en.wikipedia.org/wiki/SEER-SEM>, свободный. – Загл. с экрана.

Поступила в редколлегию 3.09.2014

**Рецензент:** д-р техн. наук, проф. И.В. Шостак, Национальный аэрокосмический университет «ХАИ», Харьков.

**ОЦІНКА ТРУДОМІСТКОСТІ РОБІТ НА РАНИХІ СТАДІЯХ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

М.А. Голованова, С.В. Надін

Публікацію присвячено питанням оцінювання трудомісткості робіт на раних стадіях створення програмного забезпечення з урахуванням великої кількості факторів в умовах обмеженості вихідної інформації. Розглянуто методи визначення трудовитрат на основі сучасних ППП, особливу увагу приділено методам з урахуванням кількості рядків коду. Запропоновано підхід, що враховує етапи життєвого циклу створення ПЗ в розрізі декомпозиції основних категорій і етапів робіт, а також враховує фактори, що впливають на трудомісткість робіт.

**Ключові слова:** програмне забезпечення, трудовитрати, обсяг робіт, методи оцінки вартості.

**LABOUR-INTENSIVENESS ESTIMATION OF WORK AT THE EARLY STAGES OF SOFTWARE CREATING**

M.A. Golovanova, E.V. Nadin

The article deals with the problems of labour-intensiveness estimation of work at the early stages of software creating taking into consideration a lot of factors in the conditions when the given information is limited. Methods of labour expenditures defining on the base of application package were considered. We paid special attention to the methods on the base of taking into consideration the number of code lines. The approach taking into account the stages of software creating life-cycle in section of main categories decomposition, stages of work and influencing factors is given.

**Keywords:** software, labour-intensiveness, amount of work, cost estimation methods.