

УДК 681.322  
DOI: 10.15827/0236-235X.119.469-473

Дата подачи статьи: 01.03.17  
2017. Т. 30. № 3. С. 469–473

## **ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА ОЦЕНИВАНИЯ ТРУДОЗАТРАТ И СТОИМОСТИ СОЗДАНИЯ ПРОГРАММНЫХ СРЕДСТВ**

*С.А. Котов, к.э.н., доцент, директор, info@gicpsvt.ru;*

*А.А. Демирский, к.т.н., начальник отдела, info@gicpsvt.ru*

*(Главный испытательный сертификационный центр безопасности программных средств и вычислительной техники, ул. Ржевская, 10, г. Тверь, 170023, Россия)*

В статье рассматривается метод оценки размера создаваемого программного средства на ранних этапах жизненного цикла его разработки. Для программных средств одного класса наибольшее влияние на трудоемкость разработки оказывает их размер. Обоснованная оценка размера создаваемого программного средства необходима для адекватной оценки трудозатрат и стоимости разработки. В статье дается краткий обзор существующих методов оценки размера создаваемого программного средства. Среди них наиболее распространенными являются метод конструктивной модели стоимости (СОСОМО, СОСОМО II) и метод функционально-балльной оценки (FPA). Указаны недостатки данных методов.

Для формирования оценки размера создаваемого программного средства предлагается использовать метод, являющийся развитием FPA и СОСОМО II. Он основан на подсчете программных объектов: экранных форм, отчетов, файлов данных (таблиц СУБД), функций импорта/экспорта данных. Для оценки размера создаваемых программных объектов используется база нечетких продукционных правил, параметры которой определяются с помощью искусственной нейронной сети (нейронная продукционная сеть Такаги–Сугено–Канга). Приводится математическое описание используемых моделей оценки. Метод был адаптирован для оценки размера создаваемого специального ПО автоматизированных систем. Выделены преимущества применения предложенного метода оценки размера создаваемого программного средства.

С использованием данного подхода была разработана информационно-аналитическая система оценивания трудозатрат и стоимости создания программных средств. В статье рассмотрена функциональная структура информационно-аналитической системы, дается перечень режимов ее функционирования.

В заключение приведены результаты применения предложенного метода для оценки размера создаваемого специального ПО автоматизированных систем.

**Ключевые слова:** оценка трудозатрат на создание программного обеспечения, управление программными проектами, нечеткие модели.

Проекты по созданию крупных программных средств (ПС) традиционно начинаются с анализа и разработки технико-экономического обоснования затрат, связанных с их созданием и развитием. Заказчику необходимо оценить объем требуемых ресурсов, а потенциальному разработчику – возможность реализации проекта в условиях и ресурсах, предлагаемых заказчиком.

Следствием низкого качества технико-экономического обоснования стоимости выполнения работ по созданию и развитию ПС зачастую являются конфликты между заказчиками и разработчиками, что отрицательно сказывается на результате.

Большей части этих негативных последствий можно избежать, используя современные методы управления проектами для их успешного завершения. При этом важное значение имеет технология формирования точных оценок затрат ресурсов, необходимых для создания и развития ПС.

При создании ПС основные затраты связаны с непосредственным или овлеществленным интеллектуальным трудом различных специалистов (аналитиков, проектировщиков, программистов и т.п.). Поэтому для измерения затрат по работам, связанным с разработкой ПС, наиболее универсальной единицей стала трудоемкость выполняемых работ, выраженная в человеко-днях или человеко-месяцах. Стоимость работ вычисляется как произведе-

ние трудозатрат на стоимость нормо-часа специалистов, выполняющих данную работу [1].

Современные методы оценки трудозатрат на создание и развитие ПС можно разделить на следующие классы:

- методы структурированной экспертной оценки;
- методы оценки по математическим моделям, построенным на основе обработки статистических данных.

Экспертные оценки применяются при отсутствии дискретных эмпирических данных. Оценки, получаемые подобным образом, представляют собой результат анализа известных эксперту проектов, в которых он принимал участие. Основным непреодолимым недостатком данной группы методов является слабое обоснование полученных оценок. Фактически единственным обоснованием оценки является авторитет эксперта (или группы экспертов).

Масштабные исследования в области математического моделирования расходов, связанных с разработкой и эксплуатацией ПС, начались в США в 1965 году. Эти разработки легли в основу нескольких примитивных моделей, созданных в конце 60–начале 70-х годов двадцатого столетия. Потом появились более сложные модели: SLIM [2], метод контрольных точек (Checkpoint) [3], PRICE-S [4],

SEER [5] и COSOMO [6]. Разработчики этих моделей оценки затрат столкнулись с одной и той же проблемой: по мере развития технологий разработки ПС происходил значительный рост размеров этого средства и увеличивалась его сложность, что существенно затрудняло точную оценку расходов на разработку [7]. Для построения математических моделей оценки в основном используется регрессионный анализ. Данную группу методов характеризует достаточно высокая точность получаемых оценок. Однако для калибровки моделей требуется большая база статистических данных по уже завершенным проектам. Кроме того, не всегда заранее известен вид зависимости входа и выхода модели.

В настоящее время наиболее распространенными методами оценки на основе математических моделей являются методы конструктивной модели стоимости (COSOMO, COSOMO II) [6] и функционально-балльной оценки (FPA) [8].

Опыт использования упомянутых выше математических моделей оценки позволяет выделить основные факторы, влияющие на затраты, связанные с созданием и развитием ПС. Для ПС одного класса наибольшее влияние на трудоемкость разработки оказывает его размер. Данный параметр может изменяться в большом диапазоне: на три-четыре порядка от  $10^4$  до  $10^7$  строк исходного кода программ [6, 8, 9]. Именно размер ПС определяет основной объем работ по проекту. Поэтому при оценке затрат на разработку ПС размер программы используется в качестве базового доминирующего параметра. Так как точность оценок размера ПС определяет точность оценок прочих стоимостных параметров проекта, методам его оценивания уделяется большое внимание. В качестве единицы измерения размера ПС обычно используется строка исходного кода (SLOC – source line of code). Данная единица измерения требует точного описания понятия «строка кода» [9]. Другая проблема использования единиц измерения размера ПС, основанных на коде, заключается в том, что на ранних стадиях жизненного цикла ПС информация об исходном коде недоступна.

Применяемые подходы для оценки размера создаваемого ПС в методах COSOMO II и FPA имеют ряд недостатков, затрудняющих их применение.

- Для использования метода функциональных баллов необходимы высококвалифицированные сертифицированные специалисты. При несоблюдении данного условия точность оценок резко снижается.

- Метод функциональных баллов требует значительного времени на сбор данных о создаваемом ПС. При этом состав собираемых данных подразумевает детальное понимание требований к ПС.

- Метод функциональных баллов ориентирован в большей степени на процедурное программирование, так как оперирует многими понятиями, не

используемыми при объектно-ориентированном подходе.

- COSOMO II использует метод оценки размера ПС с сильно ограниченным набором программных объектов. Сложность реализации каждого программного объекта определяется с помощью экспертной оценки.

- Статистике, используемой при калибровке модели COSOMO II, уже более 15 лет. Как следствие, в ней плохо учтена специфика создания ПС с использованием технологии объектно-ориентированного программирования (которая сейчас широко применяется при создании ПС).

Для формирования оценки трудозатрат на работы по созданию и развитию ПС предлагается использовать метод, являющийся развитием методов оценки размера создаваемого ПС, применяемых в FPA и COSOMO II. Это позволит частично или полностью устранить указанные недостатки. Для оценки трудозатрат на создание ПС на основе его размера могут быть использованы откалиброванные модели, применяемые в методах COSOMO II и FPA. Основной задачей является повышение точности и обоснованности оценок размера создаваемого ПС, полученных на ранних этапах жизненного цикла.

В предлагаемом методе размер ПС – это сумма оценок размеров всех программных объектов, из которых состоит ПС. Под программными объектами будем понимать конструктивные элементы ПС, на создание которых требуются значительные трудозатраты. Однотипные программные объекты можно объединить в классы. Для каждого класса разрабатывается модель для оценки размера в зависимости от значений общих характеристик (свойств) данного класса программных объектов. При использовании данного подхода процесс оценки значительно упрощается. На первом этапе аналитики занимаются определением количества и свойств создаваемых программных объектов ПС. Затем в зависимости от значений свойств этих программных объектов вычисляется их вероятный размер. Важно, чтобы в качестве свойств программных объектов выступали такие характеристики, которые могут быть доступны для оценки на начальных этапах жизненного цикла ПС. В качестве единиц измерения размера программных объектов будет использоваться SLOC. Это позволит аккумулировать оценки отдельных программных объектов для формирования итоговой оценки размера создаваемого ПС.

Для построения моделей оценки размера классов программных объектов были использованы методы искусственного интеллекта: нейронные сети и нечеткая логика (нейро-нечеткие, или гибридные модели). Особенностью этих методов является учет в них априорных сведений об объекте или о процессе исследований в форме нечетких продукционных правил «ЕСЛИ–ТО». Параметры правил

определяются по имеющимся статистическим данным с помощью алгоритма обратного распространения ошибки, генетических алгоритмов [10] и др.

Основные преимущества использования нейро-нечетких методов моделирования:

- позволяют строить модель, используя малые выборки данных (по сравнению со статистическими методами);
- не требуют предварительного определения вида зависимости входов и выходов модели;
- дают возможность моделировать сложные процессы и системы (многомерные, нелинейные);
- позволяют достаточно легко интерпретировать получаемые модели.

При использовании нейро-нечетких методов моделирования размеры экземпляров программных объектов ПС определяются с помощью *базы знаний* (БЗ), представляющей собой набор правил нечеткого вывода. Основное назначение БЗ – идентификация нелинейной зависимости  $y = \varphi(X)$ , где  $X$  – вектор значений количественных свойств класса программных объектов ПС;  $y$  – значение размера кода программного объекта ПС в SLOC.

Построение моделей оценки размера программных объектов ПС осуществляется в два этапа. На первом этапе формируется нечеткая БЗ. Данная БЗ связывает входы и выходы модели с помощью лингвистических правил «ЕСЛИ–ТО». Набор правил создается на основе извлечения знаний из имеющейся статистики. На следующем этапе производится настройка параметров нечеткой БЗ для достижения минимального отклонения модельных результатов от имеющихся статистических данных. Подобная настройка реализуется с помощью нечетких нейронных сетей, способных одновременно формировать нечеткие правила и адаптировать функции принадлежности путем модификации весов связей в процессе обучения.

Особенность применяемых моделей состоит в том, что заключения правил представляются в форме функциональных зависимостей (нейронная продукционная сеть Такаги–Сугено–Канга):

П<sub>*i*</sub>: ЕСЛИ  $x_1$  есть  $A_{i1}$  И ... И  $x_j$  есть  $A_{ij}$  И ... И  $x_m$  есть  $A_{im}$ , ТО  $y = c_{i0} + \sum_{j=1}^m c_{ij}x_j$ ,  $i = 1, \dots, n$ , где  $A_{ij}$  – нечеткое множество, отражающее значение характеристики программного объекта  $x_j$ , с функцией принадлежности  $\mu(x_j)$ ;  $x_j$  – количественная характеристика программного объекта ПС, влияющая на размер кода, который необходимо написать для его создания;  $y$  – четкая выходная переменная, равная искомому размеру кода программного объекта ПС;  $n$  – количество правил;  $m$  – количество входных переменных.

При малом объеме обучающих данных возможно использование упрощенной модели, когда заключения правил заданы константами:  $y = c_i$ ,  $i = 1, 2, \dots, n$ .

Полученная БЗ может быть интерпретирована как некоторое разбиение пространства входных характеристик на нечеткие подобласти, в каждой из которых значение выходной переменной рассчитывается как линейная комбинация входов. Переключение с одного линейного закона «входы–выход» на другой осуществляют правила БЗ. Так как границы подобластей входных переменных размыты, одновременно могут выполняться несколько правил, но с разными весами. Итоговое значение выходной переменной  $y$  определяется как суперпозиция линейных зависимостей:

$$y = \frac{\sum_{i=1}^n \left( \left( c_{i0} + \sum_{j=1}^m c_{ij}x_j \right) \prod_{j=1}^m \exp \left[ - \left( \frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right] \right)}{\sum_{i=1}^n \prod_{j=1}^m \exp \left[ - \left( \frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right]}.$$

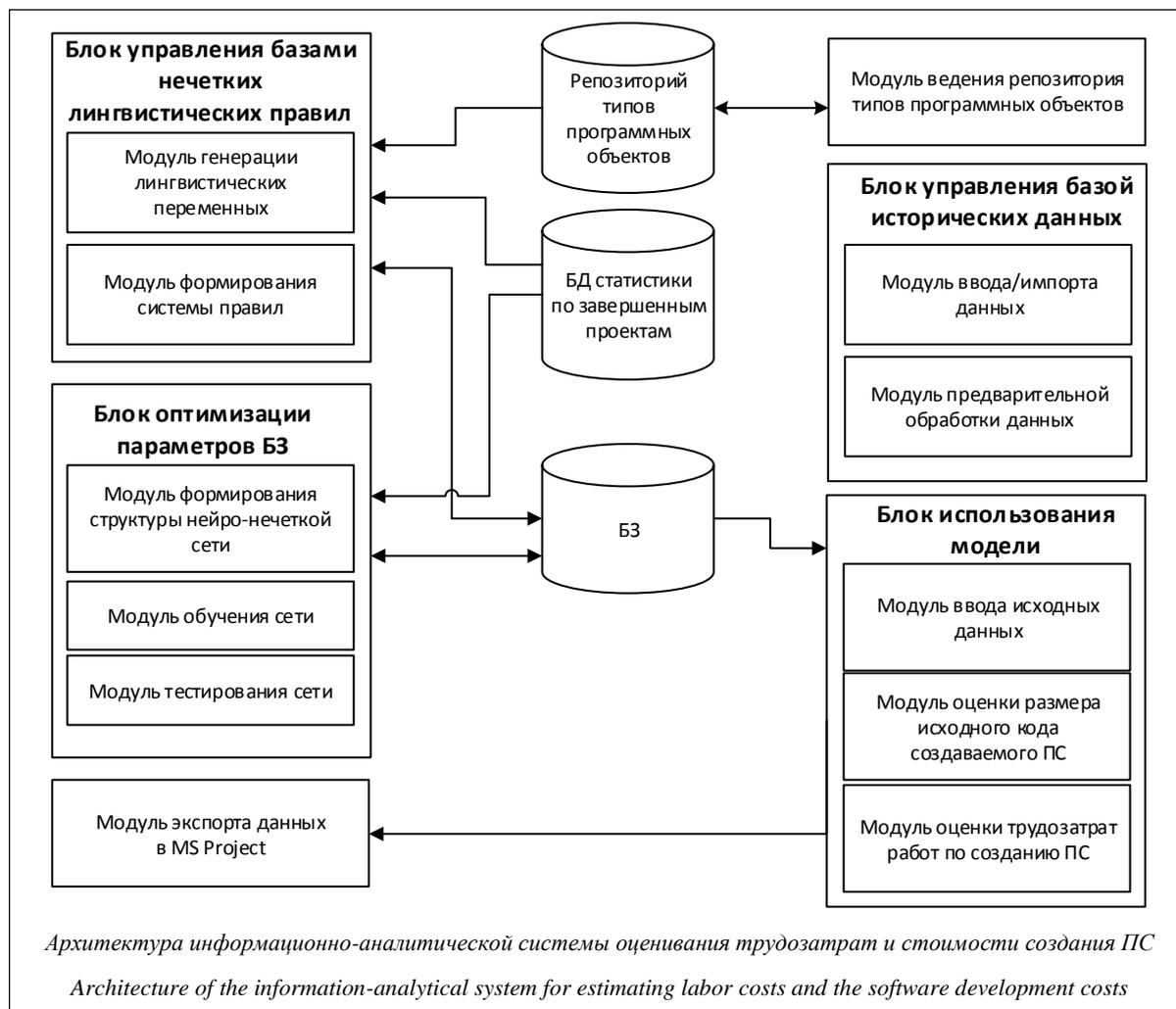
В данном выражении для задания функций принадлежности используется функция Гаусса, где  $a$  и  $b$  – параметры этой функции.

С использованием данного подхода была разработана информационно-аналитическая система оценивания трудозатрат и стоимости создания ПС. Цель такой системы – обеспечение руководителя проекта инструментарием для оперативной оценки трудозатрат на разработку ПС (как всего проекта, так и его отдельных этапов) и автоматизированной подготовки всех необходимых данных для получения этой оценки.

Информационно-аналитическая система оценивания трудозатрат и стоимости создания ПС реализует следующие функции:

- ведение БД статистики по завершенным проектам разработки ПС;
- описание структуры проекта в разрезе классов программных объектов ПС;
- выбор вида модели нечеткого вывода формируемой БЗ;
- генерация структуры лингвистических переменных, входящих в БЗ, на основе имеющейся статистики по завершенным проектам;
- генерация базы нечетких решающих правил;
- решение задачи оптимизации баз нечетких правил;
- оптимизация параметров баз нечетких решающих правил на основе имеющейся статистики по завершенным проектам;
- оценка технико-экономических параметров проекта с использованием БЗ;
- экспорт данных с результатами оценки технико-экономических параметров проекта во внешние приложения (ПО календарного планирования проектов).

Архитектура информационно-аналитической системы оценивания трудозатрат и стоимости создания ПС представлена на рисунке. По своему составу она близка к экспертным системам.



Информационно-аналитическая система оценивания трудозатрат и стоимости создания ПС может работать в одном из четырех режимов:

- ввод исходных данных для построения моделей оценки размера кода программных объектов ПС;
- построение математических моделей оценки размера кода программных объектов ПС;
- настройка (адаптация) параметров полученных моделей;
- использование сгенерированных моделей оценки размера кода программных объектов ПС для оценивания трудозатрат и стоимости их создания.

С помощью системы были построены модели оценки размера программных объектов специального ПО информационных систем. Данный класс ПС характеризуется наличием БД и глубоко проработанного графического интерфейса пользователя. В качестве типов программных объектов, характеристики которых доступны на ранних этапах жизненного цикла, выделялись следующие классы: экранные формы, диалоги, отчеты, файлы данных (таблицы СУБД), функции импорта/экспорта данных.

Разработанная информационно-аналитическая система апробирована в Главном испытательном сертификационном центре безопасности программных средств и вычислительной техники (г. Тверь). Исходные данные для формирования моделей оценки получены исходя из анализа ПС, поступивших на сертификационные испытания. Полученные результаты оценки размера ПС сравнивались с оценками, полученными по методу функциональных баллов. Были проанализированы завершённые проекты с размером ПС от  $10^4$  до  $10^6$  в строках исходного кода (SLOC). Результаты оценки размера ПС, полученные с использованием нейро-нечетких методов моделирования, отличаются от фактических на 5–10%. Данные, полученные с использованием метода функциональных баллов, отличаются от фактических на 10–23%. Оценка размера создаваемого ПС вышеперечисленными методами производилась на этапе проектирования интерфейса пользователя, после чего результат сравнивался с фактическим размером ПС после завершения его разработки.

На данный момент результаты исследования были апробированы и успешно внедрены в Банке России и Федеральной таможенной службе.

**Литература**

1. Липаев В.В. Технико-экономическое обоснование проектов сложных программных средств. М.: СИНТЕГ, 2004. 284 с.
2. Jensen Dr. Randall W., Putnam Lawrence H. Sr., Roetzheim William. Software estimating models: three viewpoints. Software Technology Support Center, 2006, pp. 23–29.
3. Jones C., Bonsignour O. The economics of software quality. Addison-Wesley, 2012, pp. 105–109.
4. Shermom D. System cost engineering. Gower Publ., 2009, 326 p.
5. Fischman Lee, McRitchie Karen, Galorath Daniel D. Inside SEER-SEM, CROSSTALK The Jour. of Defense Soft. Eng., 2005, pp. 26–28.
6. Boehm B., Abts C., Brown A.W., Chulani S., Clark B.K., Horowitz E., Madachy R., Reifer D.J., Steece B. Software Cost Estimation with COCOMO II. Englewood Cliffs, NJ, Prentice-Hall, 2000, 544 p.
7. Котов С.Л. Нормирование жизненного цикла программной продукции. М.: ЮНИТИ, 2002. 143 с.
8. Garmus D., Herron D. Functional point analysis. Addison-Wesley, 2000, 400 p.
9. McConnell S. Software estimation: demystifying the black art. Microsoft Press, 2006, 308 p.
10. Борисов В.В. Нечеткие модели и сети. М.: Горячая линия–Телеком, 2007. 284 p.

Software &amp; Systems

DOI: 10.15827/0236-235X.119.469-473

Received 01.03.17

2017, vol. 30, no. 3, pp. 469–473

**AN ENTERPRISE INFORMATION SYSTEM FOR ESTIMATING LABOR  
AND SOFTWARE DEVELOPMENT COSTS**

**S.L. Kotov**<sup>1</sup>, Ph.D. (Economics), Associate Professor, [info@gicpsvt.ru](mailto:info@gicpsvt.ru)

**A.A. Demirsky**<sup>1</sup>, Ph.D. (Engineering), Head of Department, [info@gicpsvt.ru](mailto:info@gicpsvt.ru)

<sup>1</sup> Main Testing Certification Center of Security Software and Computer Engineering, Rzevskaya St. 10, Tver, 170023, Russian Federation

**Abstract.** This article aims to study the process of software size estimation and its impact on the software development process. Software size has the greatest impact on the complexity of development. A software estimation process, which is integrated with the software development process, can help to establish realistic and credible plans to implement the project requirements and satisfy commitments. It also can support other management activities by providing accurate and timely planning information.

The article provides an overview of existing methods for estimating the size of the software created. The most common methods are: functional point analysis (FPA) and constructive cost model (COCOMO, COCOMO II). The article indicates disadvantages of these methods.

The paper presents a new approach of software size estimation. The fuzzy model Takagi-Sugeno-Kang is described by fuzzy IF-THEN rules, which represent local input-output relations of a nonlinear system. The main feature of the Takagi-Sugeno-Kang fuzzy model is to express a local dynamics of each fuzzy implication (rule) by a linear system model. The fuzzy model Takagi-Sugeno-Kang is used to estimate the size of software objects.

The paper illustrates how the new method can be used in the enterprise information system for estimating labor and software development costs. The result shows that the approach is better than FPA method.

**Keywords:** labor cost estimation, software size, software estimation, project management, fuzzy, fuzzy modeling.

**References**

1. Lipaev V.V. *Tekhniko-ekonomicheskoe obosnovanie proektov slozhnykh programmnykh sredstv* [A Feasibility Report for Complex Software Projects]. Moscow, SINTEG Publ., 2004, 284 p.
2. Jensen Dr. Randall W., Putnam Lawrence H. Sr., Roetzheim W. *Software Estimating Models: Three Viewpoints*. Software Technology Support Center, 2006, pp. 23–29.
3. Jones C., Bonsignour O. *The Economics of Software Quality*. Addison-Wesley Publ., 2012, pp. 105–109.
4. Shermom D. *System cost engineering*. Gower Publ., 2009, 326 p.
5. Fischman L., McRitchie K., Galorath D.D. Inside SEER-SEM, CROSSTALK. *The Jour. of Defense Software Engineering*. 2005, pp. 26–28.
6. Boehm B., Abts C., Winsor Brown A., Chulani S., Clark B.K., Horowitz E., Madachy R., Reifer D.J., Steece B. *Software Cost Estimation with COCOMO II*. Englewood Cliffs, NJ, Prentice-Hall Publ., 2000, 544 p.
7. Kotov S.L. *Normirovanie zhiznennogo tsikla programmnoy produktii* [Normalization of Software Products Life Cycle]. Moscow, Yuniti Publ., 2002, 143 p.
8. Garmus D., Herron D. *Functional Point Analysis*. Addison-Wesley Publ., 2000, 400 p.
9. McConnell S. *Software estimation: demystifying the black art*. Microsoft Press, 2006, 308 p.
10. Borisov V.V. *Nechetkie modeli i seti* [Fuzzy Models and Networks]. Moscow, Goryachaya liniya–Telekom Publ., 2007, 284 p.