

УДК 004.5:004.8

Дата подачи статьи: 25.08.17

DOI: 10.15827/0236-235X.121.134-139

2018. Т. 31. № 1. С. 134–139

ИСПОЛЬЗОВАНИЕ ГОРНОГО АЛГОРИТМА ДЛЯ ОЦЕНКИ ЗАТРАТ ПРИ ПРОЕКТИРОВАНИИ ПРОГРАММНЫХ ПРОЕКТОВ

Т.М. Зубкова¹, д.т.н., профессор, bars87@mail.ru

Е.Н. Наточая², к.п.н., доцент, en_ischa@mail.ru

¹ Оренбургский государственный университет, просп. Победы, 13, г. Оренбург, 460018, Россия

² Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации (Оренбургский филиал), ул. Курача, 26, г. Оренбург, 460000, Россия

Оценка затрат при разработке ПО позволяет снизить вероятность принятия неверных или непринятия нужных управленческих решений, вероятность получения незапланированных результатов программных проектов. В статье проведен анализ основных подходов к решению задачи оценки стоимости программных проектов: линейного подхода, методики функциональных точек и использования эмпирических данных. Предложена методика оценки затрат при проектировании программных проектов на основе горного алгоритма.

В статье описано математическое и программное обеспечение для решения поставленной задачи. Она выполняется по модели композиции приложения и модели раннего этапа проектирования. Для определения стоимости программного проекта предлагается зафиксировать цену на программы в соответствии с их группой сложности. Рассмотрены три группы сложности программных продуктов: простая (проектирование занимает до 4 месяцев), средняя (проектирование – от 4 месяцев до года), высокая (срок проектирования превышает год). Обосновано использование метода горной кластеризации для объединения программных продуктов в группы на основе схожести признаков.

На основе предложенной методики разработана программная система оценки стоимости программных проектов с целью повышения объективности и оперативности решений, принимаемых разработчиками. Входными данными программной системы являются структура предприятия, техническое задание, таблица оценок элементов программных продуктов, таблица Боэма. В качестве управляющего воздействия выступает алгоритм горной кластеризации. Ресурсами, поддерживающими выполнение функций программной системы, являются аппаратное и программное обеспечение предприятия. На выходе системы формируется отчет о затратах на проектирование программного продукта.

Ключевые слова: программный проект, оценка затрат, горный алгоритм, кластеризация, техническое задание, модель композиции приложения, модель раннего этапа проектирования.

Точность оценки затрат при проектировании программных проектов является важнейшим фактором, от которого зависит успешность получения качественного программного продукта (ПП) в установленные сроки и в соответствии с планируемыми бюджетом.

Следовательно, построение адекватного процесса оценки ПО позволит снизить возможность возникновения потерь, вытекающих из специфики видов человеческой деятельности, вероятность принятия неверных или непринятия нужных управленческих решений, вероятность получения незапланированных результатов программных проектов [1].

Методы оценки стоимости программных проектов широко представлены в работах зарубежных [2–4] и российских исследователей [5–7].

Можно выделить три подхода к оценке стоимости программных проектов.

1. Линейный подход. Определяет стоимость разработки как произведение количественной оценки трудозатрат и их удельной стоимости.

2. Методика функциональных точек. Основывается на том, что размер компонентов ПО оценивается в терминах количества и сложности функций, реализованных в данном программном коде.

3. Оценка с использованием эмпирических данных. Использует накопленный опыт разработки подобных проектов, макеты и прототипы програм-

мных средств (Wideband Delphi, метод ДеМарко, SLIM, COCOMO и т.д.).

Наиболее популярной моделью для оценки стоимости разработки ПО, которая практически стала стандартом, является COCOMO (Constructive cost model – конструктивная модель стоимости), разработанная Барри Боэмом. В состав усовершенствованной модели COCOMO II входят модели композиции приложения, раннего этапа проектирования и этапа постархитектуры.

Модель композиции приложений позволяет получать очень грубые величины оценки в стадии начала проекта. Она соответствует исследовательской работе, обычно выполняемой в процессе создания прототипов и анализа осуществимости проекта. Оценочное уравнение представляет собой простое линейное соотношение между объективными точками и сложностью данной области [8].

Модель ранних этапов проектирования дает более грубые оценки на той стадии жизненного цикла, на которой происходит уточнение.

В постархитектурной модели принято допущение о том, что проект имеет стабилизировавшиеся требования, планы и начальное представление о предполагаемой архитектуре. Затем проекты до конца следуют «водопадному» процессу с практически неизменными требованиями. Постархитектурная модель предназначена для получения точных оценок после того, как у проекта появляются

базовые требования, базовая архитектура и план на стадию конструирования.

Методика решения поставленной задачи

Суть первых двух моделей СОСОМО II состоит в следующем.

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. Каждый объектный экземпляр (экран, отчет) относится к одному из трех уровней сложности – простой, средний, сложный. В свою очередь, сложность является функцией от параметров клиентских и серверных таблиц данных, которые требуются для генерации экрана (делятся на три группы: меньше 3, от 3 до 7 и больше 8), а также от количества представлений и секций, входящих в отчет (делятся на три группы: от 0 до 1, от 2 до 3 и больше 4) [7].

После определения сложности количества экранов, отчетов и компонентов определяется количество объектных указателей путем перемножения исходного числа объектных экземпляров на весовые коэффициенты и последующим суммированием промежуточных результатов.

Для учета реальных условий разработки вычисляется процент повторного использования программных компонентов $\%REUSE$ и определяется количество новых объектных указателей NOP : $NOP = (\text{Объектные_указатели}) \cdot [(100 - \%REUSE)/100]$.

Для оценки затрат, основанных на величине NOP , надо знать скорость разработки продукта $PROD$. Скорость определяют с учетом уровня опытности разработчиков и зрелости среды разработки.

Проектные затраты оцениваются по формуле

$$\text{ЗАТРАТЫ} = NOP/PROD \text{ [чел.-мес.]},$$

где $PROD$ – производительность разработки, выраженная в терминах объектных указателей (значения изменяются от 4 до 50).

Модель раннего этапа проектирования. Модель используется в период, когда стабилизируются требования и определяется базисная программная архитектура.

Основное уравнение этой модели имеет следующий вид:

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

где масштабный коэффициент $A = 2,5$; показатель B отражает нелинейную зависимость затрат от размера проекта (размер системы РАЗМЕР выражается в тысячах LOC – количестве строк текста ПО); множитель поправки M_e зависит от 7 формирователей затрат, характеризующих продукт, процесс и персонал; слагаемое $\text{ЗАТРАТЫ}_{\text{auto}}$ отражает за-

траты на автоматически генерируемый программный код.

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

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

Масштабные факторы отражают следующее:

- предыдущий опыт организации в реализации проектов этого типа;
- степень гибкости процесса разработки;
- степень выполняемого анализа риска;
- насколько хорошо разработчики группы знают друг друга и насколько удачно совместно работают;
- зрелость процесса в организации.

Оценки принимают 6 значений: от очень низкой (5) до сверхвысокой (0).

Множитель поправки M_e зависит от набора формирователей затрат.

Для каждого формирователя затрат определяется оценка (по 6-балльной шкале), где 1 соответствует очень низкому значению, а 6 – сверхвысокому. На основе оценки для каждого формирователя по таблице Боэма определяется множитель затрат EM_i . К ним относятся возможности персонала, надежность и сложность продукта, требуемое повторное использование, трудность платформы, опытность персонала, средства поддержки, график работ.

Перемножение всех множителей затрат формирует множитель поправки: $M_e = \prod_{i=1}^7 EM_i$.

Слагаемое $\text{ЗАТРАТЫ}_{\text{auto}}$ используется, если некоторый процент программного кода генерируется автоматически. Поскольку производительность такой работы значительно выше, чем при ручной разработке кода, требуемые затраты вычисляются отдельно по следующей формуле:

$$\text{ЗАТРАТЫ}_{\text{auto}} = (KALOC \cdot (AT/100))/ATPROD,$$

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

Сомножитель AT в этой формуле позволяет учесть затраты на организацию взаимодействия автоматически генерируемого кода с оставшейся частью системы.

Далее затраты на автоматическую генерацию добавляются к затратам, вычисленным для кода, разработанного вручную.

Для упрощения задачи определения стоимости проекта авторы предлагают зафиксировать цену на программы в соответствии с группой сложности. ПП, проектирование которых занимает до четырех месяцев, можно отнести к простой группе сложности. Если на проектирование будет затрачено

свыше этого срока, но менее года, то ПП попадает в группу средней сложности. В группу высокой сложности попадают ПП со сроком, превышающим год.

Для объединения ПП в группы на основе схожести признаков авторы предлагают использовать метод горной кластеризации, предложенный Ягером (Yager) и Филевым (Filev) в 1993 г. Особенностью алгоритма является то, что он не требует задания количества кластеров. Выбор горного алгоритма кластеризации обуславливается его достаточно высокой скоростью выполнения и хорошей точностью.

На основании данных о затратах будем распределять ПП на три кластера: высокой сложности, средней сложности, простые.

На первом шаге горной кластеризации определяют точки, которые могут быть центрами кластеров. Для алгоритма горной кластеризации число потенциальных центров кластеров (Q) должно быть конечным. Такими центрами могут быть как объекты кластеризации (строки матрицы X), так и особые точки факторного пространства. В последнем случае диапазоны изменения входных признаков разбивают на несколько интервалов. Проведя через точки разбиения прямые, параллельные координатным осям, получаем решеточный гиперкуб. Узлы этой решетки и будут соответствовать центрам потенциальных кластеров. Обозначим через q_r количество значений, которые могут принимать центры кластеров по r -й координате, $r = 1, \dots, n$. Тогда количество возможных кластеров будет следующим: $Q = q_1, q_2, \dots, q_n$.

На втором шаге для каждой такой точки рассчитывается значение потенциала, показывающего возможность формирования кластера в ее окрестности. Чем гуще расположены объекты в окрестности прототипа кластера, тем выше значение его потенциала. Потенциал центров кластеров рассчитывается по формуле $P(Z_h) = \sum_{k=1}^M \exp(-\alpha \cdot D(Z_h, X_k))$,

где $Z_h = (z_{1,h}, z_{2,h}, \dots, z_{n,h})$ – потенциальный центр h -го кластера, $h = 1, \dots, Q$; α – положительная константа; $D(Z_h, X_k)$ – расстояние между потенциальным центром кластера (Z_h) и объектом кластеризации (X_k). В евклидовом пространстве это расстояние рассчитывается по формуле

$$D(Z_h, X_k) = \sqrt{\|Z_h - X_k\|^2}.$$

В данном случае Z – множество затрат, выраженных в человеко-часах, которые выбраны как потенциальные центры кластеров; X – простые объекты кластеризации, то есть затраты, не выбранные как потенциальные центры кластеров.

На третьем шаге итерационно выбираются центры кластеров среди точек с максимальными потенциалами – координаты «горных вершин». Центром первого кластера назначают точку с наибольшим потенциалом. Обычно наивысшая вершина

окружена несколькими достаточно высокими пиками. Назначение центром следующего кластера точки с максимальным потенциалом среди оставшихся вершин привело бы к выделению большого числа близко расположенных центров кластеров. Следовательно, перед выбором следующего кластерного центра необходимо исключить влияние только что найденного кластера. Для этого значения потенциала оставшихся возможных центров кластеров пересчитываются следующим образом: от текущих значений потенциала вычитают вклад центра только что найденного кластера (поэтому кластеризацию по этому методу иногда называют субтрактивной, от англ. subtractive – вычитание). Перерасчет потенциала происходит по формуле

$$P_2(Z_h) = P_1(Z_h) - P_1(V_1) \cdot \exp(-\beta \cdot D(Z_h, V_1)),$$

$$Z_h \neq V_1, h = 1, \dots, Q,$$

где P_1 – потенциал на 1-й итерации; P_2 – потенциал на 2-й итерации; β – положительная константа; V_1 – центр первого найденного кластера:

$$V_1 = \max(P_1(Z_1), P_1(Z_2), \dots, P_1(Z_Q)).$$

Затем снова пересчитываем значения потенциалов: $P_3(Z_h) = P_2(Z_h) - P_2(V_2) \cdot \exp(-\beta \cdot D(Z_h, V_2))$, $Z_h \neq V_1, Z_h \neq V_2, h = 1, \dots, Q$.

Итерационная процедура выделения центров кластеров продолжается до тех пор, пока максимальное значение потенциала превышает некоторый порог. Кластеризация по горному алгоритму не является нечеткой, однако ее часто используют при синтезе нечетких правил из данных [9].

Результаты реализации поставленной задачи

Рассмотренный подход к оценке затрат на разработку программных проектов очень трудоемкий, поэтому целесообразно его автоматизировать.

На рынке ПО существует целый ряд как коммерческих, так и бесплатных инструментов оценки программных проектов. Среди них наибольшей популярностью пользуются Duessa Estimate Easy UC, USC COCOMO Tool, SoftStar Systems Costar, Construx Estimate, Borland CaliberRM Estimate Professional, CostXpert.

Однако все рассмотренные инструменты не позволяют эффективно накапливать текущую информацию и выполнять анализ оценок с реальными трудозатратами и сроками по выполненным проектам на уровне организации. Поэтому на их основе сложно построить комплексную среду для оценки проектов. К настоящему моменту нет такого программного решения, которое могло бы совместить в себе оптимальность необходимой функциональности, наглядности и цены. Поэтому было принято решение о разработке собственной программной системы оценки программных проектов [10].

Начальным этапом разработки автоматизированной системы расчета затрат на программные

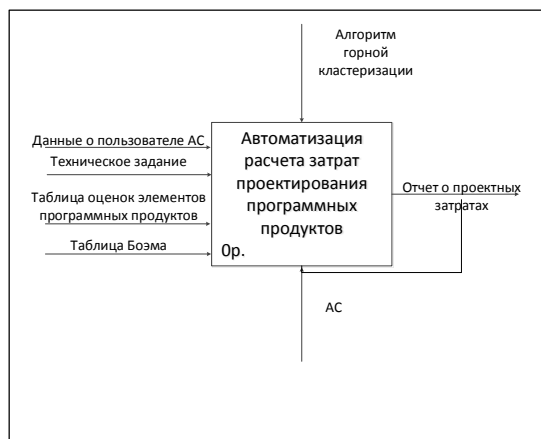


Рис. 1. Контекстная диаграмма функциональной модели автоматизированной системы

Fig. 1. Context diagram of the automated system functional model



Рис. 2. Декомпозиция 1-го уровня функциональной модели автоматизированной системы

Fig. 2. A 1st level decomposition of the automated system functional model

проекты стало создание диаграммы абстрактного уровня описания системы. Входные данные системы:

- структура предприятия: реквизиты предприятия;
- техническое задание: все данные о ПП, необходимые для расчета затрат;
- таблица оценок элементов ПП;
- таблица Бозма.

В качестве управляющего воздействия выступает алгоритм горной кластеризации.

Ресурсами, поддерживающими выполнение функций, являются аппаратное и программное обеспечение предприятия.

На выходе системы формируется отчет о затратах на проектирование ПП. Контекстная диаграмма и диаграмма первого уровня функциональной модели автоматизированной системы в нотации IDEF представлены на рисунках 1 и 2.

Для разработки программного средства были выбраны СУБД Microsoft SQL Server и Visual Studio 2012.

На рисунке 3 представлен интерфейс программного средства.

После авторизации пользователя программного средства необходимо ввести общие сведения о программном проекте, для которого рассчитываются затраты. Для этого выбираются пункты меню Данные → Справочные данные и поочередно заполняются справочники «Тип проекта», «Тип юр. лица», «Единицы измерения». Примеры заполнения справочников и введения другой информации представлены на рисунках (http://www.swsys.ru/uploaded/image/2018_1/2018-1-dop/22.jpg, http://www.swsys.ru/uploaded/image/2018_1/2018-1-dop/23.jpg).

В том же разделе вносятся данные о юридических лицах, если заказчиком или владельцем проекта является юридическое лицо. Физические лица могут быть заказчиками и владельцами проектов, а

также специалистами, работающими с данным программным проектом. Затем вводятся данные, непосредственно относящиеся к проекту (пункт меню «Проекты»). Вводится название проекта, выбираются владельцы и тип проекта из ранее введенных данных.

После внесения вспомогательной информации осуществляется переход непосредственно к расчетам. Для работы с моделью композиции приложения выбираем пункты меню Модели расчета → Модель композиции приложения. Затем выставляем все параметры проекта, нажимаем кнопку «Рассчитать» и получаем оценку затрат на проект (http://www.swsys.ru/uploaded/image/2018_1/2018-1-dop/24.jpg), кроме того, возможно сохранение данных.

Значение параметров для ввода берутся из технического задания к ПП, например, количество экранов включает экраны для ввода наименований, цены, единиц измерения т.д. (в зависимости от предметной области). Количество отчетов опреде-

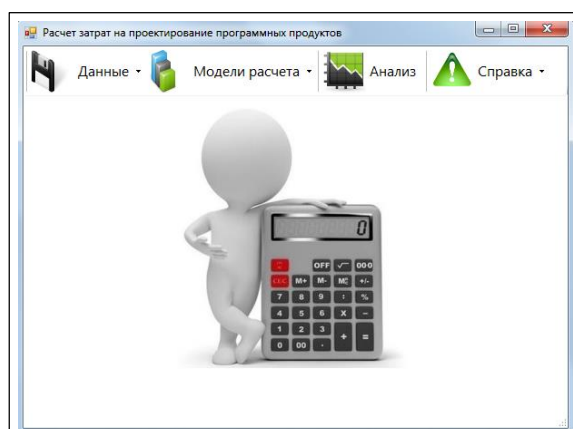


Рис. 3. Главная форма автоматизированной системы

Fig. 3. The automated system main form

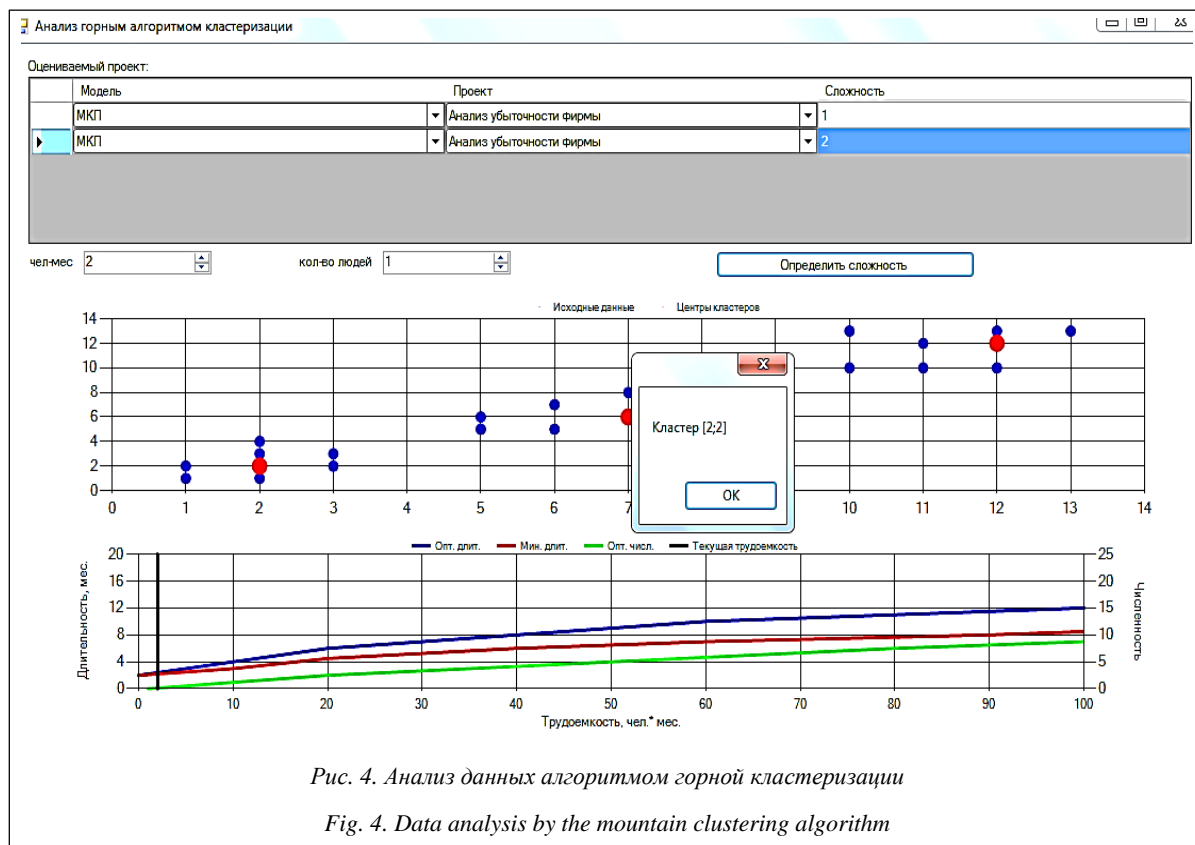


Рис. 4. Анализ данных алгоритмом горной кластеризации

Fig. 4. Data analysis by the mountain clustering algorithm

ляется также с учетом требований и функционального назначения ИП. Параметры, относящиеся к разработчикам, задаются путем оценки их профессиональных навыков и опыта работы.

Аналогичным образом выбирается модель раннего этапа проектирования, заполняются все параметры, представленные на форме, и результаты сохраняются в БД (http://www.swsys.ru/uploaded/image/2018_1/2018-1-dop/25.jpg).

Когда расчет затрат выполнен, можно переходить к анализу данных (пункт меню «Анализ»). Открывается форма, где можно выбрать проект для анализа из ранее сохраненных либо задать количество человеко-месяцев и количество персонала. После нажатия на кнопку «Определить сложность» на среднем графике отобразится группа, в которую попадает выбранный проект по сложности из трех возможных кластеров: легкий, средний или сложный. На нижнем графике по эмпирическим кривым Боэма можно определить оптимальную длительность проекта и оптимальное число персонала для него на основании полученных затрат (рис. 4).

Заключение

Таким образом, внедрение автоматизированной системы оценки стоимости проектов позволит уменьшить количество допущенных ошибок в расчетах, даст возможность учитывать большее число факторов, влияющих на стоимость и сроки выполнения проекта. Следовательно, оценка будет более

точной, что приведет к уменьшению средних сроков оформления документации, увеличению числа обслуживаемых клиентов, сокращению времени обработки и получения оперативных данных для принятия управленческих решений, к повышению степени достоверности обработки информации, степени ее защищенности, повышению эффективности хранения, обработки и анализа информации, значительному снижению трудозатрат персонала на обработку информации.

Литература

1. Зубкова Т.М., Ишакова Е.Н., Медведев А.С. Программная система оценки рисков в сфере высшего образования с использованием производственно-фреймовой модели // Вестн. ОГУ. 2014. № 1. С. 183–188.
2. Boehm B.W. Software cost estimation with COCOMO II. Prentice Hall PTR, 2000, 540 p.
3. Jones C. Software estimating rules of thumb. 2007. URL: <http://www.itmpi.org/default.aspx?pageid=197> (дата обращения: 24.08.2017).
4. Symons C.R. Function points analysis: difficulties and improvements. IEEE Transactions on Software Engineering. 1988, no. 1, pp. 2–11.
5. Архипенков С. Лекции по управлению программными проектами. URL: http://www.citforum.ru/SE/project/arkhipenkov_lectures/12.shtml (дата обращения: 24.08.2017).
6. Липаев В.В. Экономика производства программных продуктов. М.: СИНТЕГ, 2011. 352 с.
7. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения. СПб: Питер, 2012. 608 с.
8. Модель оценки стоимости COCOMO. URL: <http://project.dovidnyk.info/index.php/programnye-proekty/upravlenieproektamiposozdaniyuprogrammnogoobespecheniya/130-modelocenki-stoimostisosomo> (дата обращения: 24.08.2017).

9. Штовба С.Д. Проектирование нечетких систем средствами МАТЛАВ. М.: Горячая линия–Телеком, 2007. 288 с.
 10. Зубкова Т.М., Колобов А.Н. Расчет затрат при проекти-

ровании программных проектов по методике COCOMO II с использованием горного алгоритма. Свид. о гос. регистр. прог. для ЭВМ № 2016616453. Зарег. 10.06.2016.

Software & Systems

DOI: 10.15827/0236-235X.121.134-139

Received 25.08.17

2018, vol. 31, no. 1, pp. 134–139

COST ESTIMATION IN SOFTWARE PROJECTS DESIGN USING THE MOUNTAIN ALGORITHM

*T.M. Zubkova*¹, *Dr.Sc. (Engineering), Professor, bars87@mail.ru*

*E.N. Natochaya*², *Ph.D. (Education), Associate Professor, en_ischa@mail.ru*

¹ *Orenburg State University, Pobedy Ave. 13, Orenburg, 460018, Russian Federation*

² *Russian Presidential Academy of National Economy and Public Administration (Orenburg branch), Curacha St. 26, Orenburg, 460000, Russian Federation*

Abstract. Cost estimation in software development allows reducing the probability of making incorrect or not making necessary management decisions, the probability of obtaining unplanned results of software projects. The article performs the analysis of basic approaches to the problem of software projects cost estimation: the linear approach, the method of function points and using empirical data. The paper proposes the technique of cost estimation in software projects design based on the mountain algorithm.

The paper describes mathematical support and software to solve the problem. The task is implemented according to the application composition model and the early design model. To estimate software project costs it is offered to record the price of programs according to their group of complexity. The paper considers three groups of software products complexity: simple (design takes up to 4 months), average (design takes from 4 months to one year), high (design exceeds a year). The paper explains the use of the mountain clustering method for combining software products in groups based on a similarity of signs.

Based on the offered method, the software system of software projects cost estimation is developed in order to increase objectivity and efficiency of the decisions made by developers. Input data of the software system are a structure of an enterprise, a specification, a table of estimates of software product elements, the Boehm table. The algorithm of a mountain clustering acts as controlling influence. The resources supporting execution of software system functions are hardware and software of the company. At the system output, the report on costs of software product design is created.

Keywords: program project, cost estimation, mountain algorithm, clustering, specification, application composition model, early design stage model.

References

1. Zubkova T.M., Ishakova E.N., Medvedev A.S. Software system of risks assessment in the higher education with use of productional and frame model. *Vestn. OGU* [Vestnik of OSU]. 2014, no. 1, pp. 183–188 (in Russ.).
2. Boehm B.W. *Software cost estimation with COCOMO II*. Prentice Hall PTR Publ., 2000, 540 p.
3. Jones C. *Software Estimating Rules of Thumb*. 2007. Available at: <http://www.itmpi.org/default.aspx?pageid=197> (accessed August 24, 2017).
4. Symons C.R. Function points analysis: difficulties and improvements. *IEEE Trans. on Software Engineering*. 1988, no. 1, pp. 2–111.
5. Arkhipenkov S. *Lectures on Control of Program Projects*. Available at: http://www.citforum.ru/SE/project/arkhipenkov_lectures/12.shtml (accessed August 24, 2017).
6. Lipaev V.V. *Ekonomika proizvodstva programmykh produktov* [Software Products Production Economics]. Moscow, SINTEG Publ., 2011, 352 p.
7. Orlov S.A., Tsilker B.Ya. *Tekhnologii razrabotki programmnogo obespecheniya* [Software Development Technology]. St. Petersburg, Piter Publ., 2012, 608 p.
8. *Model otsenki stoimosti COCOMO* [The Model of COCOMO Cost Estimation]. Available at: <http://project.dovidnyk.info/index.php/programnye-proekty/upravlenieproektamiposozdaniyuprogrammnogoobespecheniya/130-modelocenkistoimostisosomo> (accessed August 24, 2017).
9. Shtovba S.D. *Proektirovanie nechetkikh sistem sredstvami MATLAB* [Fuzzy Systems Design Using MATLAB]. Moscow, Goryachaya liniya–Telekom Publ., 2007, 288 p.
10. Zubkova T.M., Kolobov A.N. *Raschet zatrat pri proektirovanii programmykh proektov po metodike COCOMO II s ispolzovaniem gornogo algoritma* [Cost Estimation in Software Projects Design by COCOMO II Method Using the Mountain Algorithm]. State registration certificate of the computer program no. 2016616453, 2016.