

5. Lutz M. (2011) Learning Python. 4th edition. Translate from English, St. Petersburg, Symbol-Plus, 1280. 6. Alchin, M. (2010). Pro Python. NY: Apress, 368. 7. Nguyen Q. (2009). CAD scripting languages: A collection of perl, ruby, python, Tcl, and SKILL scripts. San Jose CA: RAMACAD Inc, 694. 8. Schwartz R., d Foy B., Phoenix T. (2011). Learning Perl. 6th Edition. Sebastopol CA: O'Reilly Media, 390. 9. Nixon R. (2009) Learning PHP, MySQL, and JavaScript. Sebastopol CA: O'Reilly Media, 528. 10. Pratt T., Zelkovits M. (2002) Programming Languages: Design and Implementation. 4th edition. St. Petersburg, Peter, 688. 11. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. (2008) Compilers: Principles, Techniques and Tools. M., Williams, 723. 12. Hoare C. A. R. (1968) Record Handling. Programming Languages, 291 – 347.

Надійшла (received) 25.05.2014

УДК 004.412

Е. И. СОЛОВЬЕВА, студентка, ХНУРЭ, Харьков

УСОВЕРШЕНСТВОВАНИЕ МЕТОДОВ ОЦЕНИВАНИЯ ТРУДОЗАТРАТ НА СОЗДАНИЕ ИТ-ПРОЕКТА

Рассмотрены математические модели оценки трудозатрат на создание ИТ-проектов, выявлен общий вид модели оценки трудозатрат, предложен подход к повышению точности оценки трудозатрат на создание ИТ-проектов.

Ключевые слова: ИТ-проект, трудозатраты, метод функциональных точек, метод объектных точек, модель СОСОМО II.

Введение. Начиная с 2012-2013 г.г. количество ИТ-компаний на рынке Украины значительно увеличилось. ИТ-сектор улучшил свою бизнес-платформу, что позволило украинским компаниям подняться выше Индии и укрепить свой сегмент на рынке. В условиях жесткой конкуренции, ИТ-компании вынуждены повышать проходные барьеры, а также начинают разрабатывать инновационные продукты и технологии, позволяющие им занять лидирующие позиции среди известных международных игроков.

Одной из таких технологий является технология повышения точности оценивания затрат, на создание ИТ-проекта. Оценка затрат на выполнение ИТ-проекта может быть осуществлена на любом из ранних этапов создания проекта, включая инициацию. Более того, анализ затрат на начальном этапе даст возможность оценить рациональность, возможность и эффективность проекта, до того момента как документ о начале проекта будет подписан.

Анализ существующих методов расчета затрат на создание ИТ-проекта. В настоящее время существует множество методов расчета затрат на создание ИТ-проекта, среди которых можно выделить следующие [1]:

а) метод точек свойств (подобная метрика разработана, чтобы учитывать не только требования к системе, но и особенности ее реализации);

б) метод Mark II (дает одинаковый результат, как при оценке целой системы, так и при суммировании оценок полученных при анализе ее подсистем);

в) метод объектных точек (основан на положениях объектно-ориентированного подхода и модульном представлении программных систем);

г) метод ДеМарко (построен на основе использования эмпирических данных,

© Е. И. СОЛОВЬЕВА, 2014

полученные оценки корректируются с учетом хронологических данных по более ранним проектам, что дает аналитику не абстрактные показатели, а адекватные значения реальных затрат ресурсов и времени);

д) метод Wideband Delphi (метод основан на экспертных оценках по методу Delphi);

е) модели СОСОМО и СОСОМО II (модель СОСОМО в настоящее время устарела, так как подразумевала только каскадную модель жизненного цикла, и на смену ей пришла СОСОМО II, которая учитывает и спиральную и итеративную и каскадную модели, а также адаптирована к современным методологиям разработки программного обеспечения).

Однако не все методы могут использоваться на стадии инициации IT-проекта, когда разработчику необходимо определить экономическую выгодность проекта, а также необходимые для его реализации ресурсы. Методы Де-Марко, Mark II, точек свойств и Wideband Delphi не рекомендуется применять на данном этапе из-за невозможности получения необходимых оценок с приемлемым уровнем достоверности.

Поэтому, для подсчета на стадии инициации IT – проекта более рационально использовать метод функциональных точек, метод объектных точек и СОСОМО II. Использование этих методов дает аналитику приблизительное значение затрат с достаточным уровнем достоверности.

Цель работы. Метод функциональных точек на этапе инициации применим тогда, когда аналитику известны такие данные, как масштаб планируемого проекта, тип пользователей проекта, а также тип самого объекта проектирования. Данный метод позволяет найти количество функциональных точек в баллах, которые достаточно сильно отличаются от реальных значений [2].

Метод объектных точек требует аналитика точных знаний архитектуры IT-проекта, а также количество экранов, отчетов, таблиц, а также характеристику сплоченности команды, выполняющей данный проект. Данный метод достаточно точен, однако его использование оправдано лишь в случаях, когда создается типичная система, или существуют некоторые наработки [2].

Модель СОСОМО II на данный момент является самой усовершенствованной технологией, позволяющей рассчитать такие величины как время, затраченное на реализацию проекта, необходимое количество персонала, трудозатраты на IT-проект, а также необходимость в денежных средствах. Однако, одним из недостатков модели является использование показателя «тысяча условных строк кода», который не в полной степени отражает существующее положение вещей [2].

Исходя из этого, целью работы является исследование возможности симбиоза двух и более методов расчета затрат на основе взаимного преобразования метрик этих методов, отражающих знания различных участников IT-проекта о создаваемом программном продукте.

Сравнительный анализ методов расчета затрат на создание IT-проекта. Для решения поставленной задачи исследования автором в [3] предлагается подход основанный на комбинации классического метода функциональных точек, метода объектных точек и модели СОСОМО II, что позволит наиболее

полно отобразить все особенности ИС и ее программного обеспечения, которые возможно выявить в настоящее время с использованием существующих метрик. Этот подход наиболее тактически перспективен, однако для его реализации в ходе инициации ИТ-проекта следует модифицировать методы функциональных и объектных точек, чтобы они могли оценивать затраты на создание ИС на основе минимальной информации, имеющейся в распоряжении участников ИТ-проекта.

В основу классического метода функциональных точек легло представление реализуемой системы, как множества элементов, которые принадлежат к двум основным подгруппам: данные и транзакции. Данные можно разделить на два класса: внутренние логические файлы (ILF) и файлы внешних интерфейсов (EIF). Транзакции делятся на следующие классы: внешние вводы (EI), внешние выводы (EO) и внешние запросы (EQ). В классическом методе функциональных точек характеристики именно этих элементов позволяют определять количество нескорректированных функциональных точек (UFP), исходя из которого, рассчитывается количество функциональных баллов для создаваемой системы. Расчет объема продукта в функциональных баллах по методу функциональных точек осуществляется по формуле [4]

$$DFP = \left(\sum_{i=1}^{n_{ILF}} UFP_i + \sum_{j=1}^{n_{EIF}} UFP_j + \sum_{k=1}^{n_{EI}} UFP_k + \sum_{l=1}^{n_{IEO}} UFP_l + \sum_{m=1}^{n_{EQ}} UFP_m + CFP \right) \times \left((0,01 \times \sum_{p=1}^{14} DI_p) + 0,65 \right), \quad (1)$$

где DFP - количество скорректированных функциональных точек, оценивающих объем продукта (программного продукта или информационной системы), создаваемого в результате выполнения оцениваемого ИТ-проекта, с учетом

влияния общесистемных характеристик этого продукта; $\sum_{i=1}^{n_{ILF}} UFP_i$ - сумма

нескорректированных функциональных точек, характеризующая сложность структур данных, которые используются создаваемым продуктом в ходе

выполнения своих функций; n_{ILF} - количество ILF; $\sum_{j=1}^{n_{EIF}} UFP_j$ - сумма

нескорректированных функциональных точек, характеризующая сложность структур данных, которые поступают из внешних по отношению к создаваемому продукту файлов (например, из базы данных) в ходе выполнения создаваемым

продуктом своих функций; n_{EIF} - количество EIF; $\sum_{k=1}^{n_{EI}} UFP_k$ - сумма

нескорректированных функциональных точек, характеризующая сложность внешних интерфейсов для ввода данных, которые используются создаваемым продуктом для ввода данных от пользователя или устройства сбора информации

в ходе выполнения своих функций; n_{EI} - количество EI; $\sum_{l=1}^{n_{IEO}} UFP_l$ - сумма

нескорректированных функциональных точек, характеризующая сложность внешних интерфейсов для вывода данных, которые используются создаваемым

продуктом для вывода результатов выполнения своих функций; n_{EO} - количество EO; $\sum_{m=1}^{n_{EQ}} UFP_m$ - сумма нескорректированных функциональных точек, характеризующая сложность внешних запросов, которые используются создаваемым продуктом для предоставления информации пользователю или другой системе в ответ на заданные условия поиска; n_{EQ} - количество EQ; CFP - количество дополнительных функциональных точек, описывающих функции, необходимые в ходе внедрения создаваемого продукта; DI_p - системная характеристика создаваемого продукта.

Для учета общесистемных характеристик результата IT-проекта и специфических особенностей команды исполнителей IT-проекта результаты расчета нескорректированных функциональных точек могут умножаться на дополнительный коэффициент. Таким образом, выражение (1) может быть приведено к виду $Y = AX + B$, где B равен 0.

Рассмотрим подход к вычислению нескорректированных функциональных точек как элементов выражения (1). Основным способом определения значений UFP для любых ILF и EIF является оценка их сложности, которая в общем случае может быть описана следующим образом [4]:

$$UFP_i = f(\langle \sum_r^{n_{RET}} RET_r, \sum_s^{n_{DET_F}} DET_{F_s} \rangle), i = 1, \dots, n_{ILF};$$

$$UFP_j = f(\langle \sum_r^{n_{RET}} RET_r, \sum_s^{n_{DET_F}} DET_{F_s} \rangle), j = 1, \dots, n_{EIF},$$
(2)

где RET – сущность (Record Element Type), описывающая какой-либо объект предметной области автоматизируемого объекта или процесса, или же отдельный аспект этого объекта; n_{RET} - количество RET в каждом конкретном ILF или EIF; DET_F – уникальный атрибут (Data Element Type), используемый для описания упомянутой выше сущности; n_{DET_F} - количество DET_F в каждом конкретном ILF или EIF.

В общем случае RET может состоять из одного или нескольких DET_F , однако каждый DET_F должен принадлежать одному RET .

Основным способом определения значений UFP для любых EI, EO и EQ является оценка их сложности, которая в общем случае может быть описана следующим образом [4]:

$$EI_k = f(\langle \sum_r^{n_{FTR}} FTR_r, \sum_s^{n_{DET_T}} DET_{T_s} \rangle), k = 1, \dots, n_{EI};$$

$$EO_l = f(\langle \sum_r^{n_{FTR}} FTR_r, \sum_s^{n_{DET_T}} DET_{T_s} \rangle), l = 1, \dots, n_{EO};$$

$$EQ_m = f(\langle \sum_r^{n_{FTR}} FTR_r, \sum_s^{n_{DET_T}} DET_{T_s} \rangle), m = 1, \dots, n_{EQ},$$
(3)

где FTR – количество ILF и/или EIF, которые модифицируются или считываются в ходе реализации транзакции (File Type Referenced); n_{FTR} - количество FTR в

каждом конкретном EI, EO и EQ; DET_T – уникальный элемент, используемый для реализации транзакции; n_{DET_T} – количество DET_T в каждом конкретном EI, EO и EQ.

Метод объектных точек рассчитывает оценку объема предполагаемого к созданию программного кода в процессах инициации IT-проектов создания программных продуктов. Этот метод более прост, чем классический метод функциональных точек и дает немного более точные результаты. Каждая объектная точка может относиться к одной из следующих групп: экраны, отчеты и 3GL-модули создаваемого программного продукта. Расчет объема продукта в объектных точках осуществляется по формуле [5]:

$$NOP = \left(\sum_{i=1}^{n_{SCR}} OP_i + \sum_{j=1}^{n_{REP}} OP_j + \sum_{k=1}^{n_{3GL}} OP_k \right) \times (100 - \%reuse) / 100, \quad (4)$$

где NOP – количество новых объектных точек, оценивающее объем создаваемого продукта в объектных баллах; $\sum_{i=1}^{n_{SCR}} OP_i$ – сумма объектных точек, характеризующая

сложность программных объектов типа «экран», используемых для ввода данных в создаваемый продукт и вывода данных пользователю в ходе выполнения функций продукта; n_{SCR} – количество программных объектов типа «экран»;

$\sum_{j=1}^{n_{REP}} OP_j$ – сумма объектных точек, характеризующая сложность программных

объектов типа «отчет», используемых для вывода данных в виде бумажных или электронных документов пользователю в ходе выполнения функций создаваемого продукта; n_{REP} – количество программных объектов типа «отчет»;

$\sum_{k=1}^{n_{3GL}} OP_k$ – сумма объектных точек, характеризующая сложность программных

объектов типа «3GL-модуль», используемых для обработки данных в создаваемом продукте в ходе выполнения функций продукта; n_{3GL} – количество программных объектов типа «3GL-модуль»; $\%reuse$ – процент повторно используемых программных объектов указанных типов, величина которого определяется экспертным путем.

Таким образом, выражение (3) также можно представить как частный случай модели вида $Y = AX + B$. Отличие от метода классического функциональных точек заключается, главным образом, в точках зрения на элементы анализируемой ИС, что выражается в подходе к расчету значений объектных точек как меры сложности отдельных программных объектов создаваемого продукта [5]:

$$OP_i = f \left(\left\langle \left\langle \sum_r^{n_{SRVR}} SRVR_r, \sum_s^{n_{CLNT}} CLNT_s \right\rangle, \sum_t^{n_{Views}} Views_t \right\rangle \right), i = 1, \dots, n_{SCR}; \quad (5)$$

$$OP_i = f \left(\left\langle \left\langle \sum_r^{n_{SRVR}} SRVR_r, \sum_s^{n_{CLNT}} CLNT_s \right\rangle, \sum_t^{n_{VSections}} Sections_t \right\rangle \right), i = 1, \dots, n_{REP},$$

где $SRVR$ – таблица данных, расположенная на сервере и используемая в сочетании с экраном или отчетом; $CLNT$ – таблица данных, расположенная на

клиентской станции и используемая в сочетании с экраном или отчетом; *Views* - используемое представление, составляющее оцениваемый экран; *Sections* - используемая секция, составляющая оцениваемый отчет.

Количество объектных точек для 3GL-модулей в методе объектных точек по умолчанию предполагается равным 10.

Для оценивания трудозатрат в ходе планирования IT-проекта рекомендуется использовать модель COCOMO II Early Design, которая имеет следующий вид [5]:

$$PM = A \times [Size']^B \times \prod_{i=1}^7 EM_i + PM_M, \quad (6)$$

где PM - прогнозируемое значение трудозатрат на выполнение иницируемого IT-проекта; A - поправочный коэффициент модели COCOMO II, $A = 2,5$; $Size'$ - количество строк кода иницируемого IT-проекта; B - коэффициент, отражающий влияние на трудозатраты иницируемого IT-проекта масштаба и экономичности этого проекта; EM_i - драйвер затрат IT-проекта, $i = 1, \dots, 7$; PM_M - значение трудозатрат на реинжиниринг и конверсию программного кода, предполагаемые к выполнению в рамках иницируемого IT-проекта.

Данная модель позволяет измерить объем создаваемой системы с помощью показателя имеющего реальное физическое воплощение. Поэтому, как показывает практика, модель COCOMO II в большинстве случаев позволяет получить наиболее точные результаты трудозатрат и затрат времени на создание системы.

Для повышения точности расчет затрат авторы модели COCOMO II вводят в модели специальные поправочные коэффициенты. Однако, в отличие от метода функциональных точек, где все системные характеристики сведены в одну группу, модель COCOMO II позволяет разделить поправочные коэффициенты на следующие три группы [5]:

а) общесистемные коэффициенты, отражающие технологическую специфику разработки программных модулей командой исполнителей проекта (AA, AT, SU, UNFM, DM, CM и IM);

б) драйвера затрат проекта (RCPX, RUSE, PDIF, PERS, PREX, FCIL и SCED);

в) драйвера масштаба и экономичности проекта (PREC, FLEX, RESL, TEAM и PMAT).

Обсуждение результатов. В ходе инициации IT-проекта создания ИС информация о сущностях и атрибутах предметной области может быть получена в ходе предпроектного обследования автоматизируемого объекта или процесса, однако информацию о предполагаемых к использованию программных элементах получить затруднительно.

Использование метода объектных точек в ходе инициации IT-проекта также связано с определенными затруднениями. Так, информация о схеме базы данных ИС в ходе инициации IT-проекта создания ИС не может быть получена. Информацию о предполагаемом разделении экранов и отчетов можно получить в ходе предпроектного обследования документов, действующих на автоматизируемом объекте или процессе, исходя из предположения о переносе

структуры существующих документов в планируемые к созданию экраны и отчеты.

В качестве элементарных показателей, позволяющих осуществить расчет затрат на создание IT-проекта, модель COCOMO II использует величины *KNSLOC*, *KASLOC* и *BRAC*. Эти величины являются количественными и имеют материальное воплощение в виде конкретных строк кода, написанных «с нуля» или адаптированных, а также процента кода, написанного и выброшенного из проекта из-за изменения требований к ИС и ее функциям.

Одним из основных различий между методом объектных точек и моделью COCOMO II является то, что модель COCOMO II не делает различия между типами и назначением программных компонентов, воспринимая их все как различные варианты программных модулей, а в методе функциональных точек разница между экраном, отчетом и 3GL-компонентом важна.

Результатом объединения классического метода функциональных точек и метода объектных точек, можно считать представление ИС, которое:

а) четко определяет границы системы

б) делает акцент на тип объекта, который находится на границе системы, между ней и пользователем и является либо экраном, либо отчетом, а также учитывает наличие 3GL-модулей, которые отвечают за выполнение функций системы.

в) учитывает потоки данных, которые циркулируют внутри системы, а также между ней и другими приложениями (базы данных, и т.д.).

Представление ИС с точки зрения комбинации метода объектных точек и классического метода функциональных точек показано на рис. 1.

UFP(EI) являются транзакциями внешнего ввода, который пользователь осуществляет непосредственно в интерфейсы экранов. Экраны состоят из

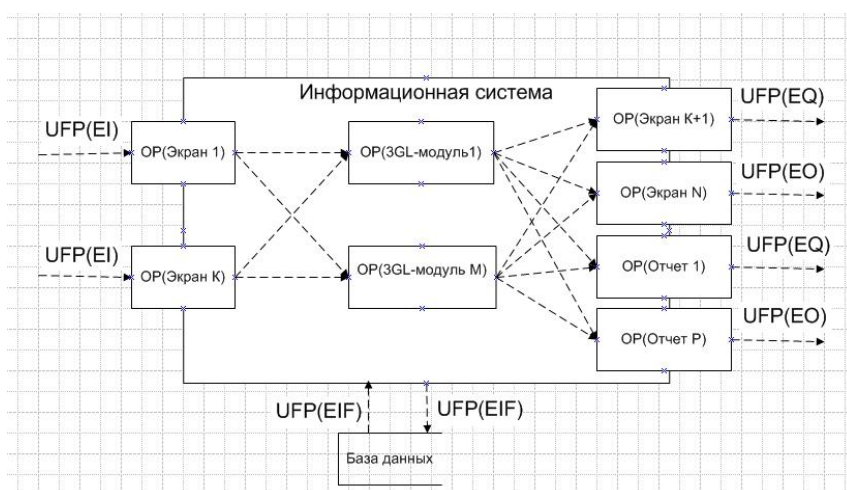


Рис. 1 – Представление информационной системы с точки зрения комбинации метода объектных точек и классического метода функциональных точек

определенного количества представлений (Views) и таблиц баз данных, с уже известной структурой. Далее транзакции отправляются на выполнение к 3GL-модулям, которые являются основой системы и представляют собой внутренние логические файлы (ILF). При обращении к БД запрос выделяет необходимые данные, а именно файлы внешних интерфейсов (EIF). Далее результат в виде внешнего вывода (EO) или внешнего запроса (EQ) подается на выход системы как экран, либо отчет, также с уже известной структурой.

Предлагаемая комбинация методов функциональных и объектных точек

позволяет вносить уточнения в модель, описывающую объем продукта, являющегося результатом иницируемого IT-проекта. Эти уточнения определяются как:

а) процент повторно используемых объектных точек, отражающий объем решений по информационному и программному обеспечению, повторно используемый в ходе создания IT-проекта;

б) множество системных характеристик DI_p , $p = 1, \dots, 14$, отражающее общесистемные особенности результата создаваемого IT-проекта.

Такой подход позволяет описать объем работ по созданию ИС как результата выполнения IT-проекта с помощью показателя «количество объектно-функциональных точек». В общем случае этот показатель будет рассчитываться по формуле

$$DOFP = \left(\sum_{i=1}^{n_{ILF}} UFP_i + \sum_{j=1}^{n_{EIF}} UFP_j + NOP + CFP \right) \times \left((0,01 \times \sum_{p=1}^{14} DI_p) + 0,65 \right), \quad (7)$$

где $DOFP$ - количество объектно-функциональных точек, характеризующих ту часть результата выполнения IT-проекта, которую следует создавать «с нуля».

В общем случае значение NOP рассчитывается по формуле (4), однако для определения количества объектных точек выражение (5) следует модифицировать таким образом:

$$OP_i = f \left(\left\langle \left\langle \sum_r^{n_{FTR}} FTR_r, \sum_s^{n_{DET}} DET_{T_s}, \sum_t^{n_{Views}} Views_t \right\rangle \right\rangle, i = 1, \dots, n_{SCR}; \right) \quad (8)$$

$$OP_i = f \left(\left\langle \left\langle \sum_r^{n_{FTR}} FTR_r, \sum_s^{n_{DET}} DET_{T_s}, \sum_t^{n_{Sections}} Sections_t \right\rangle \right\rangle, i = 1, \dots, n_{REP}, \right)$$

Подобный способ определения количества объектных точек позволяет отказаться от необходимости использовать информацию о базе данных ИС и разрешит проводить оценку, основываясь исключительно на представлениях структур данных и командных элементов, присутствующих на экранах и отчетах как программных объектах ИС.

Выражение (7) позволит также отказаться от представления 3GL-модулей как элементов ИС, количество объектных точек которых остается неизменным вне зависимости от особенностей структур данных, обрабатываемых в этих модулях. Это повысит точность расчета объема работ по созданию ИС и, соответственно, повысит точность оценки трудозатрат на создание соответствующего IT-проекта.

Предлагаемая комбинация методов функциональных и объектных точек может быть использована в модели СОСОМО II при условии, что каждый программный объект будет рассматриваться как самостоятельный программный модуль, для реализации которого следует написать определенное количество строк программного кода. В этом случае количество строк кода может быть получено на основе комбинации методов функциональных и объектных точек одним из следующих способов [6]:

а) в результате применения метода отката (backfiring);

б) в результате статистического анализа количеств строк кода, реализующих аналогичные модули в ранее выполненных IT-проектах.

Выводы. Предлагаемая комбинация методов функциональных и объектных точек с последующим использованием результатов расчета в модели СОСОМО II позволяет повысить точность оценивания затрат на создание IT-проекта в ходе его инициации следующим образом:

а) за счет максимального использования доступных в ходе инициации IT-проекта сведений о предметной области и создаваемом продукте;

б) за счет использования возможности уточнения модели расчета затрат путем ввода дополнительных коэффициентов (системные характеристики классического метода функциональных точек);

в) предоставления возможности использования модели СОСОМО II в ходе инициации IT-проекта, позволяющей учитывать особенности проекта, участников проекта и технологий его выполнения.

Список литературы: 1. Шафер, Д. Управление программными проектами: достижение оптимального качества при минимуме затрат [Текст] / Д. Шафер, Р. Фатрелл, Л. Шафер. – М.: Издательский дом «Вильямс», 2003. – 1136 с. 2. Евланов, М. В. Задача оценивания затрат на создание информационной системы [Текст] / М. В. Евланов, Е. И. Соловьева // *Materialy X mezinardni vedecko – prakticka conference «Veda a vznik – 2013/2014»*. – Dil 34. *Moderni informacni technologie*. – Praha: Publishing House «Education and Science», 2014. – С. 45-48. 3. Соловьева, Е. И. Подход к повышению точности оценивания трудозатрат на создание IT-проекта [Текст] / Е. И. Соловьева // *Радиоелектронні і комп'ютерні системи*. – 2014. - № 1 (65). – С. 144-149. 4. *Functional Point Counting Practices Manual. Release 4.1.1* [Text]. – Troy: IFPLUG, 2001. – 370 p. 5. *COCOMO II Model Definition Manual* [Электронный ресурс] // Сайт «Center for Systems and Software Engineering». – Режим доступа: ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf. 6. Соловьева, Е. И. Сервис «Анализа затрат на создание IT-проекта информационной системы» [Текст] / Е. И. Соловьева // 18-й Международный молодежный форум «Радиоэлектроника и молодежь в XXI веке». Сб. материалов форума. Т. 6. – Харьков: ХНУРЭ, 2014. – С. 149-150.

Bibliography (transliterated): 1. *Shafer, D, Fatrell, R., Shafer, L.* (2001). *Software project management: achieving optimal quality at the lowest cost*. Moskva: Publising house “Vilyams”, 1136 p. 2. *Ievlanov, M. V., Solovyova, E. I.* (2014). The problem of estimating the cost of an information system. *Materialy X mezinardni vedecko–prakticka conference «Veda a vznik – 2013/2014»*, Dil 34. *Moderni informacni technologie*, Praha, Publishing House «Education and Science», 2014, 45-48. 3. *Solovyova, E. I.* (2014). The approach to improve the accuracy of estimation of labor costs for the creation of an IT-project. *Radioelectronics and computer systems*. Kharkov: Publishing house KhNAU “KhAP”, № 1 (65), 144-149. 4. *Functional Point Counting Practices Manual. Release 4.1.1* [Text]. Troy: IFPLUG, 2001, 370. 5. *COCOMO II Model Definition Manual* [Electronic Resource] // «Center for Systems and Software Engineering», ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf. 6. *Solovyova, E. I.* (2014). Service “Cost analysis to establish IT-project of information system”. 18 International youth forum “Radioelectronics and youth in XXI century”, Vol. 6, Kharkov, Publishing house KhNURE, 149-150.

Поступила (received) 17.05.2014