

УДК 614.2
МРНТИ 20.15.05

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К ОЦЕНКЕ СТОИМОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

С.К. САГНАЕВА¹, Г.К. СЕМБИНА², У.М. СМАЙЛОВА³

¹ЕНУ им. Л.Н. Гумилева

²Международный университет информационных технологий

³Центр педагогического мастерства

Аннотация: Первым шагом в расчете трудозатрат разработчика является оценка размера программного обеспечения информационной системы. Разработка программного обеспечения всегда связана с некоторыми рисками и неопределенностью. В настоящее время нет единой методологии для оценки стоимости разработки программного обеспечения. В современных методах практически не представлены формализованные методы классификации и подсчета трудоемкости разработки программной системы. Основное различие в методе, используемом для оценки затрат на рабочую силу, это тип используемых критериев оценки качества (количественный или качественный). Сама процедура определения размера программного обеспечения может отличаться методиками расчетов. Приведенный метод CETIN соответствует основным ИТ-стандартам, этапам жизненного цикла ПО. В рамках модели технологической зрелости СММ, рассматриваемой в рамках ИТ-проекта, для оценки трудозатрат была использована технология Rational Unified Process (RUP), обеспечивающая итерационный и инкрементный (наращиваемый) подход к созданию ПО, а также планирование и управление проектом на основе функциональных требований к системе. В работе предложены простые в использовании показатели, которые допускают корректировку и автоматизацию расчетов трудозатрат на разработку прикладного программного обеспечения. Для учета нефункциональных требований, таких как отказоустойчивость, восстанавливаемость, безопасность введены частные поправочные коэффициенты.

Ключевые слова: программное обеспечение, информационная система, метод, стоимость, методология, модель, коэффициент

OBJECT ORIENTED APPROACH TO THE SOFTWARE DEVELOPMENT COST ESTIMATION

Abstract: The first step in calculating the developer's work is to estimate the size of the information system software. Software development is always associated with certain risks and consequences. The formalized methods of classification and software development effort estimation are practically not presented in traditional methods.

There is currently no uniform methodology for estimating the cost of software development. The main difference in the method used to estimate labor costs is the type of quality assessment criteria used (quantitative or qualitative). The procedure for determining the size of the software may differ by calculation methods. Presented CETIN method is compliance with both core IT standards and life cycles stages of software. The Rational Unified Process (RUP) technology was used, as part of the technological complexity of the SMM model, to estimate the labor costs, to provide an iterative and incremental approach for software development, as well as for planning and managing the projects based on the functional requirements of the system. The paper proposed easy-to-use indicators that allow the adjustment and automation of labour cost calculations for developing software application. Special correction factors are proposed to take into account non-functional requirements, such as fault tolerance, recoverability, safety.

Keywords: software, information system, method, cost, methodology, model, coefficient

БАҒДАРЛАМАЛЫҚ ҚАМТАМАСЫЗ ЕТУДІ ҚҰРУ ҚҰНЫН ЕСЕПТЕУДІҢ ОБЪЕКТІЛІ-БАҒЫТТАЛҒАН ТӘСІЛІ

Аңдатпа: Ақпараттық жүйе құрғанда оған жұмсалатын шығын мөлшерін есептеу бағдарламалық қамтаманың көлемін бағалаудан басталады. Бағдарламалық қамтамасыз етуді әзірлеу әрқашан кейбір тәуекелдермен және белгісіздікпен байланысты. Қазіргі заманауи әдістерде бағдарламалық жүйені әзірлеудің еңбек сыйымдылығын жіктеу мен есептеудің формальды әдістері іс жүзінде көрсетілмеген. Жұмыс күшіне арналған шығындарды бағалау үшін пайдаланылатын әдістегі негізгі айырмашылық – сапаны бағалаудың пайдаланылатын өлшемдерінің (сандық немесе сапалық) түрі. Бағдарламалық қамтамасыз ету мөлшерін анықтау процедурасының өзі есептеу әдістемелерімен ерекшеленуі мүмкін. Келтірілген SETIN әдісі негізгі АТ стандарттарына, бағдарламалық қамтамасыз етудің өмірлік циклінің кезеңдеріне сәйкес келеді. Ақпараттық технологиялар жобасының аясында қарастырылатын СММ технологиялық жетілу моделі шеңберінде еңбек шығындарын бағалау үшін бағдарламалық қамтама құруға итерациялық және инкременттік (ұлғайтылатын) тәсілді қамтамасыз ететін Rational Unified Process (RUP) технологиясы, сондай-ақ жүйеге қойылатын функционалдық талаптар негізінде жобаны жоспарлау және басқару қолданылды. Бас тарту тұрақтылығы, қалпына келу, қауіпсіздік сияқты функционалды емес талаптарды есепке алу үшін жеке түзету коэффициенттері енгізілді.

Түйінді сөздер: бағдарламалық қамтама, ақпараттық жүйе, әдіс, құны, әдіснама, модель, коэффициент

Введение

В разработке программного обеспечения одной из самых сложных задач является определение временных и стоимостных затрат на разработку нового программного продукта. С беглого взгляда, кажется, что проблема проста, однако то, что кажется простой проблемой на поверхности, на самом деле гораздо технически сложнее. Прежде всего, программные продукты имеют некоторую функциональность, которая имеет обыкновение изменяться, становится некоторой переменной величиной, что влияет на изменение стоимости проекта. Другим фактором, влияющим на стоимость проекта, является время. Как только фактор времени становится переменной величиной, то теряется контроль над стоимостью проекта. Поэтому актуальность рассматриваемого вопроса связана с необходимостью рассмотрения проблем, связанных с оценкой затрат при разработке новых программных продуктов.

Исследования отдельных авторов [1] показывают, что оценка трудоемкости проектов необходима, прежде всего, для решения таких проблем как повышение общей точности оценки бюджета проекта, анализа риска проекта и выбора оптимального решения,

планирования и управления проектом, анализа стоимости улучшения качества программных проектов. Необходима адекватная и достоверная оценка стоимости программного обеспечения. Известные к настоящему моменту методики оценки трудозатрат плохо адаптированы к реальному процессу проектирования ПО. Для некоторых проектов крупных информационных систем (ИС) данные методики практически неприменимы. По данным Software Engineering Institute (SEI) около 80% всех внедренных систем количественной оценки процесса разработки программного обеспечения (ПО) оказываются практически не востребуемыми на протяжении первых двух лет.

Исторически известны различные методы оценки трудоемкости, которые достаточно сложны, причем сложность оценки трудоемкости увеличивалась с увеличением размера программных продуктов.

Значительно исследование моделирования стоимости программного обеспечения началось с обширного исследования SDC в 1965 году из 104 атрибутов программных проектов [Nelson 1966] (Э.А.Нельсон, сотрудник компании SDC) [2]. Это привело к некоторым

частично полезным моделям в конце 1960-х и начале 1970-х годов. В конце 1970-х годов появились модели, такие как SLIM [Putnam and Myers 1992], Checkpoint [Jones 1997], PRICE-S [Park 1988], SEER [Jensen 1983] и COCOMO [Boehm 1981]. Хотя большинство из этих исследователей приступили к разработке моделей стоимостной оценки примерно в одно и то же время, все они столкнулись с одной и той же дилеммой: по мере роста и значимости программного обеспечения, оценка также возрастала по сложности, что затрудняло точное определение прогноза стоимости разработки ПО. Эта динамическая область оценки программного обеспечения поддерживала интересы этих исследователей, которым удалось установить основные этапы затрат на разработку программного обеспечения [4].

В конце 1970-х годов сотрудник IBM Алан Альбрехт предложил концепцию функциональной точки (FPA) на основе оценки количества функциональных требований, предъявляемых клиентами [4]. В дальнейшем, были разработаны методы точек свойств, FPA Mark II, трехмерных функциональных точек и метод объектных точек.

Следующий этап связан с появлением модели COCOMO (конструктивной модели затрат), разработанной американским инженером Барри Бемом в начале 1980-х годов [5]. Он основан на подсчете строк кода. Обновленная версия модели появилась в 1995 году.

В странах СНГ использовались методы, предложенные ведущими учеными и разработчиками программного обеспечения РФ: В. Л. Арлазаровым [6], В. В. Липаевым [7] и другими. В Республике Казахстан до 2012 года была использована методология, разработанная ОАО «НИТ». Компанией ТОО КСИ Фактор (г.Астана) была предложена методика оценки трудоемкости командной разработки программного обеспечения (ПО) для информационных систем (2011) [8]. Однако с появлением объектно-ориентированных методов сущность разработки программного обеспечения изменилась и потребовались новые подходы к исследованию вопросов оценки трудоемкости разработки ПО.

В работе будет рассмотрена методика CETIN, предложенная ТОО «КСИ Фактор», и являющаяся основой методики расчета затрат на создание, развитие и сопровождение информационных систем государственных органов [9].

1. Универсальная модель жизненного цикла RUP

Одним из самых популярных методов оценки и улучшения процессов можно назвать модель технологической зрелости (Capability Maturity Model, CMM). При использовании CMM модель процесса оценивается по шкале из 5 уровней зрелости процесса (начальный уровень, повторяемые процессы, стандартные процессы, управляемые процессы, оптимизируемые процессы).

CMM применима к оценке в таких областях как: разработка программного обеспечения, системная инженерия, управление проектами, управление рисками, информационные технологии (ИТ) и управление персоналом. Это обусловлено тем, что процессы создания и сопровождения ПО последовательно совершенствуются и выявляются дополнительные возможности используемых процессов: анализируются дефекты и выявляются причины их возникновения, новые технологии стандартизируются и адаптируются в технологическую систему в рамках всей организации. Процессы создания ПО оцениваются в целях предотвращения ситуаций, приводящих к дефектам, а результаты оценки учитываются в последующих проектах.

Технологические требования рассматриваются только до 3 уровня. Поэтому для достижения минимум третьего уровня CMM рекомендуется использование технологии Rational Unified Process (RUP) [10]. RUP обеспечивает итерационный и инкрементный (наращиваемый) подход к созданию ПО, планирование и управление проектом на основе функциональных требований к системе - вариантов использования, построение системы на базе архитектуры ПО.

Согласно RUP, жизненный цикл (ЖЦ) ПО делится на отдельные циклы, в каждом из

которых образуется новое поколение продукта. Любой цикл, в свою очередь, делится на четыре последовательные стадии: начальную (inception), стадию разработки (elaboration), стадию конструирования (construction) и стадию ввода в действие (transition).

Любая стадия исчерпывается в четко поставленной контрольной точке (milestone). В данный промежуток времени должны достигаться имеющие большое значение результаты и учитываться критически важные решения о последующей разработке.

В рамках RUP определены шесть важнейших дисциплин:

- бизнес-моделирование (business modeling) – определение необходимых средств системы и нужд пользователя;
- управление требованиями к системе (requirements) – развертывание общей идеи системы совокупно с функциональными и нефункциональными условиями ее работы;
- анализ (analysis) и проектирование (design) – представление способов исполнения системы на этапе реализации;
- реализация (implementation) кодирование и генерация действующих программных модулей системы;
- тестирование (test) – диагностика функционирования системы;
- внедрение (employment) – обеспечение системы конечным пользователям и их просвещение.

Указанные этапы будут выделены в расчетах по методике CETIN.

2. Методологии описания требований к системе

В рамках первого этапа RUP необходимо проведение процесса бизнес-моделирования, который реализуется в рамках различных методик, в зависимости от выбранного подхода к моделированию: структурного/функционального или объектного. Функциональный подход рассматривает организацию как набор функций, преобразующий входной поток информации в выходной поток. Особенностью является четкое отделение функций от самих данных. При объектном подходе рассматрива-

ваемая предметная область есть некоторый набор взаимодействующих объектов (предметов или явлений), имеющих четко определяемое поведение.

Каждый из подходов обладает своими преимуществами. Функциональный подход лучше воспринимается при обследовании предметной области при получении информации от исполнителей об их обязанностях. Объектный подход позволяет построить более устойчивую к изменениям систему, лучше соответствует существующим структурам организации.

Важным качеством объектного подхода является согласованность моделей деятельности организации и моделей проектируемой информационной системы от стадии формирования требований до стадии реализации. По объектным моделям может быть прослежено отображение реальных сущностей моделируемой предметной области (организации) в объекты и классы информационной системы.

Большинство существующих методов объектно-ориентированного подхода включают язык моделирования и описание процесса моделирования. В качестве языка моделирования объектного подхода используется унифицированный язык моделирования UML, который содержит стандартный набор диаграмм для моделирования [11]. На этапе управления требованиями к системе язык UML позволяет проводить визуализацию, специфицирование, конструирование и документирование объектно-ориентированных систем.

UML-модель имеет два основных аспекта: семантическую информацию (семантику) и визуальное представление (нотацию). С точки зрения семантики модели, программное приложение — это набор взаимосвязанных логических конструкций, таких, например, как классы, ассоциации, состояния, варианты использования и сообщения. Семантические элементы определяют содержание модели. Семантика используется для создания программного кода, контроля правильности модели, измерения ее сложности и т. д.

Нотация — это визуальное представление семантики модели. Визуальное представ-

ление дает людям возможность непосредственно работать с моделью, то есть просматривать и редактировать ее. В рамках исследуемой предметной области выделяются действующие лица/акторы (actor) и описываются их варианты использования (use case).

Используются диаграммы вариантов использования (use case diagram), классов (class diagram), коммуникаций (communication diagram) и диаграмма узлов (node diagram).

3. Функциональный размер системы в рамках методики SETIN

Функциональный размер ИС в методике задается набором из пяти элементов, каждый элемент которого измеряется в соответствующей функциональной единице измерения. Наименования и обозначения функциональных единиц измерения:

C-количество вариантов использования (Case);

E-количество типов объектов (бизнес объектов) (Entity);

T-количество свойств типов объектов (Tool);

I- количество взаимодействий между типами объектов (Interaction);

N-количество типов узлов (Node).

Количество вариантов использования (C) - оценивается подсчетом количества вариантов использования информационной системы, изображенных на диаграмме вариантов использования. Количество типов объектов (E) оценивается подсчетом неодинаковых классов, изображенных на диаграмме классов. Количество свойств типов объектов (T) оценивается подсчетом количества свойств (атрибутов) типов объектов, изображенных на диаграмме классов. Количество взаимодействий между типами объектов (I) оценивается подсчетом количества связей (отношений) между классами на диаграмме коммуникаций. Количество узлов (N) оценивается подсчетом

количества типов узлов на диаграмме узлов.

Базовая трудоемкость S_j процесса разработки в рамках RUP с номером $j=1, 2, \dots, 6$ рассчитывается по следующей формуле [16]:

$$S_j = 1/165 \cdot [C \cdot S_j(C) + E \cdot S_j(E) + T \cdot S_j(T) + I \cdot S_j(I) + N \cdot S_j(N)], \quad (1)$$

где

S_j - трудоемкость процесса разработки с номером j в [человеко-месяц];

J - номер процесса разработки;

$S_j(C)$ - нормативные коэффициенты трудоемкости реализации одного варианта использования в процессе разработки с номером j {[человеко-час]/[вариант]};

$S_j(E)$ - нормативный коэффициент трудоемкости реализации одного типа объектов в процессе разработки с номером j {[человеко-час]/[тип объектов]};

$S_j(T)$ - нормативный коэффициент трудоемкости реализации одного свойства типа объекта в процессе разработки с номером j {[человеко-час]/[свойство типа объектов]};

$S_j(I)$ - нормативный коэффициент трудоемкости реализации одного взаимодействия между типами объектов в процессе разработки с номером j {[человеко-час]/[взаимодействие между типами объектов]};

$S_j(N)$ - нормативный коэффициент трудоемкости реализации одного типа узла в процессе разработки с номером j {[человеко-час]/[узел]};

$SIZE=\{C,E,T,I,N\}$ - функциональный размер информационной системы, в функциональных единицах измерения

165 - количество человеко-часов в одном человеко-месяце.

Нормативные коэффициенты, используемые в формуле (1), приведены в таблице 1 и их значения были установлены путем экспертной оценки.

Таблица 1- Нормативы трудоемкости по процессам в разрезе функциональных единиц

№	Наименование процесса	Функциональная единица измерения				
		C	E	T	I	N
1	Бизнес-моделирование	32,12	28,33	0	14,15	0
2	Управление требованиями	58,03	28,04	0	20,32	0
3	Проектирование	45,42	61,75	31,35	37,52	24,02
4	Реализация	31,57	81,51	50,72	36,11	0
5	Тестирование	88,96	0	0	0	0
6	Сопровождение	8,69	0	0	0	23,74

3.1 Расчет базовой трудоемкости системы

Допустим, что в ходе обследования установлен функциональный размер системы:

$$SIZE = \{C, E, T, I, N\} = \{55, 6, 32, 18, 6\}.$$

Базовая трудоемкость разработки ПО определяется на основе оценки трудоемкости каждого процесса разработки ПО, которая есть сумма произведений единиц измерения функционального размера и значений нормативных коэффициентов трудоемкости соответственно формуле (1). Расчеты базовой трудоемкости для указанных выше 5 процессов были автоматизированы в соответствии с рисунком 1, 2.

3.2 Расчет трудоемкости с учетом поправочных коэффициентов

В методике предусмотрено использование 18 частных поправочных коэффициентов, учитывающих требования к системе. Например, для коэффициента «Режим эксплуатации ИС», который определяется в зависимости от конкретных технологий или типов обработки, принятых в системе, предусмотрены значения: «1» -обработка данных в режиме реаль-

Данные	Базовая трудоемкость	Скорректированная трудоемкость	Стоимость
Значения нормативных коэффициентов расхода разработчика			
№	Наименование показателя	Обозначение	Норматив
1	Накладные расходы	Пнр	71.5%
2	Расходы периода	Прп	48%
3	Рентабельность	Пр	25%
4	Козфф.э.ласт трудоемкости	L	0.75%
5	Козфф.трудоемкости сопр.ППО	N	15%

Нормативы трудоемкости по процессам в разрезе функциональных единиц						
Функциональная единица измерения						
Трудоемкость, чел.час						
№	Название процесса	Вариант исп	Тип объекта	Св. типа объекта	Св. отн. объектов	Тип узла
1	Бизнес моделиние	32,12	28,33	0,00	14,15	0,00
2	Управл. требования	58,03	28,04	0,00	20,32	0,00
3	Проектирование	45,42	61,75	31,35	37,52	24,02
4	Реализация	31,57	81,51	50,72	36,11	0,00
5	Тестирование	88,96	0,00	0,00	0,00	0,00
	Развертывание	88,69	0,00	0,00	0,00	23,74

Рис. 1 – Значения нормативных коэффициентов

Данные	Базовая трудоемкость	Скорректированная трудоемкость	Стоимость
Нормативы затрат			
Количество вариантов использования C:		55	
Количество типов объектов E:		6	
Количество свойств типов объектов T:		32	
Количество взаимодействий между типами объектов I:		18	
Количество узлов N:		6	
Посчитать			
Процессы разработки			
Бизнес моделирование:	S1	13,2804848484848	
Управление требованиями:	S2	22,5796969696969	
Проектирование:	S3	28,432	
Реализация:	S4	27,2632121212121	
Тестирование:	S5	29,6533333333333	
Развертывание:	S6	3,75993939393939	
Базовая трудоемкость:		124,968666666667	

Рис. 2 – Расчет базовой трудоемкости по процессам

ного времени, «1,04» - параллельная обработка данных, «1,05» - обработка данных в режиме реального времени, «1,07» - совмещенная обработка данных.

Частные коэффициенты выбираются пользователями самостоятельно в зависимости от типа разрабатываемого ПО ИС. Выбранные показатели отображены в таблице 1.

Поправочные коэффициенты для соответствующих процессов разработки вычисляются по формулам:

$$КП1 = K11 \cdot K16 \cdot K17 \quad (2)$$

$$КП2 = K1 \cdot K2 \cdot K4 \cdot K5 \cdot K6 \cdot K7 \cdot K8 \cdot K9 \cdot K16 \cdot K17 \cdot K18 \quad (3)$$

$$КП3 = K1 \cdot K2 \cdot K4 \cdot K5 \cdot K6 \cdot K7 \cdot K8 \cdot K9 \cdot K11 \cdot K12 \cdot K13 \cdot K14 \cdot K15 \cdot K16 \cdot K17 \cdot K18 \quad (4)$$

$$КП4 = K1 \cdot K2 \cdot K4 \cdot K5 \cdot K6 \cdot K7 \cdot K8 \cdot K9 \cdot K10 \cdot K12 \cdot K13 \cdot K14 \cdot K15 \cdot K16 \cdot K17 \cdot K18 \quad (5)$$

$$КП5 = K1 \cdot K2 \cdot K4 \cdot K5 \cdot K6 \cdot K7 \cdot K8 \cdot K9 \cdot K10 \cdot K11 \cdot K12 \cdot K13 \cdot K14 \cdot K15 \cdot K16 \cdot K17 \cdot K18 \quad (6)$$

$$КП6 = K1 \cdot K2 \cdot K11 \cdot K16 \cdot K18 \quad (7)$$

Скорректированную трудоемкость разработки прикладного ПО (ППО) рассчитывают по формуле:

$$S = КП1 \cdot S1 + КП2 \cdot S2 + КП3 \cdot S3 + КП4 \cdot S4 + КП5 \cdot S5 + КП6 \cdot S6 \quad (8)$$

Расчетные показатели поправочных коэффициентов для каждого процесса разработки ПО получены в соответствии с рисунком 3.

При расчете стоимости необходимо учитывать такие показатели, как «Средне-

месячная номинальная заработная плата» и «Средний размер инфляции». При помощи этих данных рассчитывается результирующее значение в соответствии с рисунком 4. Стоимость разработки прикладного ПО составит 38 033 490,37 тг. При предположении, что средняя заработная плата программиста составляет 250 т.тг.

Для определения времени разработки ПО необходимо учесть показатель скоррек-

тированной трудоемкости разработки ПО, равный 149,55 чел.-мес. В методике указана таблица «Зависимость срока разработки от трудоемкости», в которой необходимо найти нижнюю и верхнюю границы сроков разработки ПО. Для полученного нами значения нижняя граница равна 9 месяцам, верхняя – 11. Берем среднее значение, равное 10 мес. Таким образом, рассчитана стоимость разработки ППО и срок его разработки.

Таблица 1 - Значения частных поправочных коэффициентов

Показатель	Обозначение	Значение
1.Режим эксплуатации ИС (выбран режим «обработка данных в режиме разделения времени»)	K1	1
2. Масштаб ИС (выбран масштаб «средние ИС - от 11 до 100 пользователей с длительным ЖЦ с возможностью роста до крупных систем»)	K2	0,99
3. Стабильность ИС («постоянное внесение изменений»)	K3	1,1
4. Защита от несанкционированного доступа (выбран уровень «средняя»)	K4	1
5. Защита программ и данных (выбран уровень «средняя»)	K5	1
6. Контрольный след операций (возможность фиксации несанкционированных изменений в системе, выбран режим «Выборочное отслеживание»)	K6	1,08
7. Отказоустойчивость (выбран режим «средняя»)	K7	1
8. Восстанавливаемость (выбран уровень «средняя»)	K8	1
9. Длительность обработки/время отклика (выбран уровень «умеренная»)	K9	1
10. Исходный язык разработки ИС (выбран показатель «объектно-ориентированный»)	K10	1
11. Класс пользователя (выбран показатель «средний»)	K11	1,07
12. Требования к центральному обрабатывающему устройству (процессору) (выбран показатель «средний»)	K12	1
13. Требования к оперативной (основной) памяти (выбран показатель «большая»)	K13	1
14. Требования к внешней памяти (выбран показатель «малая»)	K14	1,01
15. Требования к локальной вычислительной сети (выбран показатель «высокие требования»)	K15	1
16. Критичность ИС (выбран показатель «организационная безопасность»)	K16	1
17. Готовность (выбран показатель «запатентованное (методика разработчика»)	K17	1,09
18. Представление данных (выбран показатель «объектный»)	K18	1

Оценка стоимости разработки ППО ИС по методике СЕТИН

Данные | Базовая трудоемкость | Скорректированная трудоемкость | Стоимость

Поправочный коэффициент

Режим эксплуатации ИС К1:	1 обработка данны	Исходный язык разработки ИС К10:	1 объектно-орие
Масштаб ИС К2:	0.99	Класс пользователя К11:	1.07 средний
Стабильность ИС К3:	1.1	Требования к центральному обр устр К12:	1 средняя
Защита от несанкц. доступа К4:	1 средняя	Требования к оперативной памяти К13:	1 большая
Защита программ и данных К5:	1 средняя	Требования к внешней памяти К14:	1 большой
Контрольный след операций К6:	1 не имеется	Требования к локальной вычислительной сети К15:	1 высокие требо
Отказоустойчивость К7:	1 средняя	Критичность К16:	1 организационн
Восстанавливаемость К8:	1 средняя	Готовность К17:	1.09 запатентов
Длительность обработки К9:	1 умеренная	Представление данныт К18:	1 реляционный

Посчитать

Поправочные коэффициенты разработки и сопровождения ППО

Поправочный коэффициент 1	1.1663
Поправочный коэффициент 2	1.24700796
Поправочный коэффициент 3	1.24700796
Поправочный коэффициент 4	1.17708228
Поправочный коэффициент 5	1.15548444
Поправочный коэффициент 6	1.0593

Скорректированная трудоемкость: 149.5456685

Рис. 3 – Расчет скорректированной трудоемкости

Данные | Базовая трудоемкость | Скорректированная трудоемкость | Стоимость

Стоимость

Зарплата:	250000
Среднемесячная номинальная заработная плата:	250151.666666667
Средний размер инфляций:	0.060666666666667

Оценка стоимости разработки ППО ИС:

Средняя стоимость 1 человека-месяца:	54326.927327384
Стоимость разработки ППО:	38033490.37

Рис. 4 – Пример расчета стоимости разработки ППО

Заключение

Стоимость программного обеспечения – это оценка усилий, необходимых для разработки и сопровождения ПО. Известно, что средняя стоимость проекта составляет от первоначальной оценки 178% для крупных компаний, 182% для средних предприятий и 214% для малых и средних предприятий.

История развития методов оценки показывает, что практически ни один из методов не является совершенным, поскольку существует множество факторов, влияющих на оценку. Большинство компаний используют свой исторический опыт, нежели хорошо известные методы. Кроме того, некоторые из проектов, которые были завершены за пределами указанных сроков или стоимости проекта, частично были уменьшены за счет функции, указанной в исходных условиях. Если

эта ситуация не предусмотрена контрактом с клиентом, это приведет к дополнительной потере репутации компании-разработчика.

Как правило, при разработке сложных программных систем, входящих в корпоративную информационную систему, необходимо снизить зависимость качества от результатов субъективных факторов, таких как квалификация, опыт, организация процесса разработки и т. д. Для этого требуется разработка промышленных технологий для оценки ресурсов, необходимых для разработки ПО.

Методика CETIN может быть использована для получения адекватной оценки стоимости разработки ПО, поскольку проводится в рамках RUP и учитывает все этапы создания прикладного ПО.

ЛИТЕРАТУРА

1. Nelson, E.A., (1966), "Management Handbook for the Estimation of Computer Programming Costs", Systems Development Corp.
2. Wolverton, R. W, (1974), "The cost of developing large-scale software", IEEE Transactions on Computer, vol.23(6), P. 615–636.
3. Albrecht, A.J., (1979), "Measuring Application Development Productivity", Proceedings Share/Guide IBM Applications Development Symposium, Monterey, CA
4. Barry W. Boehm, Chris Abts, Sunita Chulani. Software Development Cost Estimation Approaches - A Survey. University of Southern California, IBM Research. (<http://sunset.usc.edu/publications/TECHRPTS/2000/usccse2000%505/usccse2000%505.pdf>)
5. Arlazarov, V. L., Slavin, O. A., Shustov, A. V., (2007), "Estimation of the value of the information-technical complex of a complex system", Proceedings of the ISA RAS, vol. 29, P.152-182
6. Lipaev, V.V., (2004), "The feasibility study of complex software projects", Sinteg, Moscow, Russia
7. Vendrov, A.M., (2016), "Software design of economic information systems", Finance and Statistics, Moscow, Russia
8. Gabbasov, MB, Pustovoitenko, V.V., Ualieva, A.A., (2011), "Evaluation of the development time of information systems based on teamwork", Materials of the scientific-practical conference "The use of mathematical modeling and information technology in the study of socio-economic problems" dedicated to the 50th anniversary of Ph.D., associate professor Gabbasova MB, NII EITT, Astana, Kazakhstan, P.160- 163
9. Minister of Investment and Development of the Republic of Kazakhstan (2016), No. 133, "The method of calculating the cost of creating, developing and maintaining information systems of state bodies", Astana, Kazakhstan
10. Kotchen, Philip, (2002), "Introduction to the Rational Unified Process", Williams, Moscow, Russia
11. Fowler, Martin, (2006), "UML Distilled", Symbol Plus, Moscow, Russia