Настоящие методические указания содержат рекомендации по структуре и содержанию лабораторных работ по технико-экономическому обоснованию программных проектов студентов специальностей “Программная инженерия” и “Компьютерные науки”, связанных с разработкой программных продуктов (информационных технологий, систем, баз данных и т.д.) различного назначения.

Учитывая, что программный продукт может быть создан как под конкретного заказчика, так и для рыночного распространения необходимо обосновать перед заказчиком стоимость (договорную цену) и согласовать вопросы авторского сопровождения разработки - **в первом** случае, либо определить рыночную цену программного продукта и минимальный объем продаж (для обеспечения безубыточной работы организации) - во **втором** случае.

Все расчеты должны сопровождаться соответствующими пояснениями, ссылками на источники получения исходных данных. Формулы должны приводиться с расшифровкой условных обозначений. В расчетах необходимо использовать текущие рыночные цены и тарифы на продукцию, работы, услуги, действующие на момент разработки проекта, курсы иностранных валют для пересчета валютной выручки и цен в иностранной валюте.

1. Теоретическая часть

Программное обеспечение должно быть представлено, как объект продажи и поставки и ориентировано на конкретного покупателя. Потенциальные покупатели должны иметь возможность сравнивать свои требования к программному продукту с характеристиками, описанными в документе, проводить анализ конкурентных преимуществ в сравнении с другими программными системами.

1.1. Описание программного продукта

Описание программного продукта должно соответствовать стандартам в области информационных технологий ГОСТ ИСО/МЭК 12119-2000 «Информационная технология. Пакеты программ. Требование к качеству и тестирование», в котором даны соответствующие рекомендации по структуре документа «Описание программного продукта» [1] и ГОСТ 28195-89 «Оценка качества программных средств. Общие положения», в котором изложены разделы надежность, практичность, эффективность, сопровождаемость, мобильность [2].

Описание программного продукта должно содержать следующие основные разделы.

Обозначения и указания:

- обозначение описания продукта (обозначение продукта должно включать наименование продукта и обозначение его версии или даты выпуска);
- рабочая задача (возможности программного продукта, доступные конечному пользователю);
- соответствие нормативным документам;
- необходимая программно-аппаратная платформа;
- интерфейсы с другими продуктами;
- объекты поставки (определяется каждый физический компонент поставляемого продукта, в частности техническая документация и все носители данных, в обязательном порядке необходимо установить вид поставляемых программ, например исходные программы, объектные (рабочие) модули или загрузочные модули);
- ввод в действие (инсталляция) - указывается, будет ли инсталляция продукта проводиться пользователем, или нет;
- поддержка (предлагается ли поддержка при эксплуатации продукта);
- сопровождение (предлагается ли сопровождение продукта).

Функциональные возможности:

- пригодность и согласованность (приводится набор функций, реализуемых в данной версии и соответствие этих функций стандартам, соглашениям, методическим рекомендациям и др.);
- граничные значения (минимальные или максимальные значения; длины ключей; максимальное число записей в файле; максимальное число критериев поиска; минимальный объем выборки и т.д.);
- защита (технология предотвращения случайного или преднамеренного несанкционированного доступа к программам и данным).

Надежность – способность программного продукта в конкретных областях применения выполнять заданные функции в соответствии с программными документами в условиях возникновения отклонений в среде функционирования, вызванных сбоями технических средств, ошибками во входных данных, ошибками обслуживания и другими дестабилизирующими воздействиями (приводится набор программных и организационных средств, обеспечивающих возможность сохранять качество функционирования программного продукта при установленных условиях за установленный период времени (проверка достоверности исходных данных; зашита против серьезных последствий ошибки пользователя; восстановление работоспособности программной системы при ошибках)).

Практичность – свойства, способствующие быстрому освоению, применению и эксплуатации программного продукта с минимальными трудозатратами с учетом характера решаемых задач и требований к квалификации обслуживающего персонала. Характеризуется набором атрибутов, относящихся к объему затрат, требуемых для использования программного продукта определенным кругом пользователей, в том числе:

- интерфейс пользователя (тип интерфейса: командная строка; меню; окна, функциональные клавиши; функция подсказки);
- требуемая квалификация пользователя (набор знания необходимых пользователю для эксплуатации программного продукта: знание соответствующей технической области; знание операционной системы; знания, получаемые в результате специального обучения и т.д.);
- адаптация к требованиям пользователя (если продукт может настраиваться пользователем, то должны быть указаны инструментальные средства для проведения такой настройки и условия их применения: изменение параметров, алгоритмов вычислений; назначение функциональных клавиш и т.д.);
- защита от нарушения авторских прав (техническая защита от копирования; запрограммированные даты окончания использования продукта; интерактивные напоминания об оплате за копии и пр.);

Эффективность – степень удовлетворения потребности пользователя в обработке данных с учетом экономических, вычислительных и людских ресурсов. Приводиться информация о характере поведения продукта, например времена ответа и оценки производительности при реализации заданных функций при установленных условиях - для заданных конфигураций системы и профилей загрузки и др.

Сопровождаемость - характеризует технологические аспекты, обеспечивающие простоту устранения ошибок в программе и программных документах и поддержания в актуальном состоянии:

- возможность диагностики в случае отказов;
- определение условий для модификации, либо изменения режимов эксплуатации;
- возможность тестирования модифицированных частей программного продукта.

Мобильность - возможность адаптации продукта к различным условиям эксплуатации без применения дополнительных сервисов, простота внедрения в новых условиях; взаимозаменяемость с другим (аналогичным) программным продуктом.

Примечание: разделы *надежность, практичность, эффективность, сопровождаемость, мобильность* должны соответствовать ГОСТ 28195-89 “Оценка качества программных средств. Общие положения” [3]. Пример описания программного продукта приведен в Приложении 1.

1.2. Технико-экономическое обоснование договорной цены

Под технико-экономическим обоснованием стоимости (договорной цены) программной системы будем понимать методику оценивания трудовых, временных и финансовых ресурсов по созданию программной системы, соответствующей требованиям заказчика. В основу определения требуемых объемов ресурсов должны быть положены:

- совокупность функций (бизнес-процессов), реализуемых в будущей программной системе и их относительная важность (приоритет) для заказчика;
- требования к функциональной полноте и качеству реализации каждой функции.

В качестве основных показателей оценки стоимости программной системы используются:

- сложность (размеры) программной системы;

- трудозатраты на разработку;
- длительность разработки программной системы в целом и ее отдельных этапов;
- численность и квалификация специалистов, привлекаемых к созданию программной системы;
- фонд оплаты труда специалистов на создание программной системы в целом и по конкретному этапу жизненного цикла;
- прочие прямые затраты и накладные расходы, связанные с созданием программной системы.

В основу определения размеров программной системы положено понятие «сложности», под которой понимается количество элементов программной системы (программных компонент, файлов, входных и выходных документов) и взаимосвязей между ними.

Под термином трудозатраты понимается суммарный объем труда специалистов для создания программного продукта. В качестве универсального измерителя трудозатрат используется показатель – человеко-месяц. Каждый человеко-месяц содержит 160 человеко-часов (четыре недели, пять рабочих дней, восьмичасовой рабочий день).

Длительность разработки и численность специалистов определяются на основе трудозатрат и нормативной производительности труда программиста, выражаемой в количестве строк кода, создаваемых программистом в единицу времени. При этом выделяются следующие этапы жизненного цикла программной системы:
- анализ требований к программной системе;
- определение спецификаций;
- проектирование;
- кодирование;
- тестирование и комплексные испытания;
- опытная эксплуатация.

В реализации проекта на каждом этапе принимает участие три группы специалистов:
- руководитель проекта, системные аналитики;
- непосредственные разработчики программных систем и специалисты по комплексированию;
- технический персонал, обеспечивающий тестирование, документирование и опытную эксплуатацию программного обеспечения.

Фонд оплаты труда на реализацию проекта определяется исходя из производительности труда специалиста и согласованной базовой месячной заработной платы.

Все нормативы и другие статистические данные, используемые в методиках технико-экономического обоснования стоимости основываются на статистических данных, обобщающих зарубежный и российский опыт разработки программных систем [3].

1.2.1 Требования к программным системам

По уровню сложности все множество программных систем (**ПС**) можно разбить на три типа. К первому типу относятся:
- комплексные программные системы (**КПС**) и технологии, отдельные части которых реализованы на различных платформах;
- территориально-распределенные программные системы и технологии;
- системы автоматизированного либо автоматического управления, функционирующие в режиме реального времени.

Второй тип составляют программные информационно-справочные системы (**ИПС**), обеспечивающие информационную поддержку основных функций (бизнес-процессов) организации с большим количеством типов исходной информации.

К третьему типу относятся инженерные и научно-технические пакеты программ (**ППП**) и технологий, характеризующихся четко заданным алгоритмом обработки и малыми объемами исходных данных.

При формировании требований к программной системе необходимо использование стандарта ISO/IEC 9126-1:2001 [4]. С учетом этого, системы первого и второго типа должны удовлетворять следующим требованиям:
- архитектура системы должна соответствовать текущим и перспективным целям и стратегическим, функциональным задачам создаваемой системы;

- в структуре и компонентах следует предусматривать обеспечение максимально возможной сохранности капитальных вложений заказчика в аппаратные и программные средства, а также в базы данных при длительном развитии, сопровождении и модернизации системы;
- для обеспечения перспективы развития системы должны быть предусмотрены возможность интеграции компонентов и мобильность ПС на различные аппаратные и операционные платформы на основе концепции и стандартов открытых систем;
- архитектура информационной системы должна быть достаточно гибкой и допускать простое, без коренных структурных изменений, развитие и наращивание функций и ресурсов системы в соответствии с расширением сфер и задач ее применения;
- необходимо обеспечить эффективное использование ресурсов системы и минимизировать интегральные затраты на обработку данных в типовых режимах ее функционирования с учетом текущих эксплуатационных затрат и капитальных вложений в создание **ПС**;
- следует обеспечить комфортный, максимально упрощенный доступ конечных пользователей к управлению и результатам функционирования системы на основе современных графических средств и наглядных пользовательских интерфейсов.

1.2.2 Методы оценки трудозатрат на разработку программной системы

1.2.2.1 Прямой метод оценки трудозатрат

Прямой метод определения технико-экономических параметров программной системы основан на использовании метода экспертных оценок. Программная система декомпозируется до уровня элементарных компонент. Оценка размеров каждого из компонент производиться либо внешним экспертом, имеющим опыт разработки подобных систем и готовые прототипы, либо специалистами разработчика и заказчика [5].

При декомпозиции следует использовать градации, термины и определения соответствующие ГОСТ 34.003-90 [6]. Основные из них следующие:
- интегрированная программная система – совокупность двух и более программных систем, в которых функционирование одной из них зависит от результатов функционирования другой;
- программная система – совокупность программных комплексов, реализующих множество функций организации;
- программный комплекс – совокупность программных компонент, реализующих конкретную функцию (бизнес-процесс);
- сложная программная компонента – совокупность программных кодов, реализующих сложную функцию бизнес-процесса;
- программная компонента – совокупность программных кодов, реализующих элементарную функцию бизнес-процесса.

Размеры программной системы определяются в виде количества строк исходного кода в терминах «Lines of code-**LOC**» [3]. При оценке количества строк исходного кода следует учитывать следующие положения:
- строка исходного кода содержит только один оператор;
- определение (описание) исходных данных учитывается один раз;
- не учитываются строки, содержащие комментарии и отладочные операторы;
- учитывается каждая инициализация, вызов, либо включение макроса в качестве исходного кода.

В качестве базового показателя количества строк исходного кода следует использовать число операторов языка ассемблер. Преобразование размеров программы по этому измерителю в размеры программы кода, написанного на других языках программирования и наоборот, представлены в таблице 1.

Каждый из экспертов должен дать оптимистическую *o*, пессимистическую *p* и реалистическую *r* оценки (табл. 2).

Средняя оценка по компонента программной системы $\left(e_{ij}^{k}\right)$ бета-распределению будет определяться как:

$$e_{ij}^{k} = \frac{\left(o_{ij}^{k} + 4r_{ij}^{k} + p_{ij}^{k}\right)}{6} \qquad (1)$$

где *k* - индекс (номер) эксперта;

i - индекс (номер) программного комплекса;
j - индекс (номер) программного компонента.

Таблица 1

Показатель **LOC** для языков и систем программирования

№	Язык программирования	Ассемблер LOC	Показатель LOC на функциональную точку
1	Basic Assembler	1	320
2	Macro Assembler	1,5	213
3	Basic	3	107
4	Pascal	3,5	91
5	C++	6	53
6	Java	6	53
7	Oracle, Sybase	8	40
8	Access	8,5	38
9	Delphi	11	29
10	Oracle Developer/2000	14	23
11	Smalltalk	15	21
12	Cobra	16	20
13	HTML 3,0	22	15
14	SQL (ANSI)	25	13
15	Excel	50	6

Таблица 2

Бланк экспертной оценки размерности программной системы.
(Язык программирования _____)

Состав программной системы	Оценки		
	Оптимистическая	Реалистическая	Пессимистическая
1. Программная система			
1.1. Программный комплекс			
1.1.1. Программная компонента 1			
1.1.2. Программная компонента 2			
……			
……			

Результирующая оценка размера программ определяется путем суммирование экспертных оценок:

$$R = \frac{\sum_{k=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M} e_{ij}^{k}}{L} \qquad (2)$$

где *L* - количество экспертов;
N - число программных комплексов в программной системе;
i M - число программных компонентов в *i* программном комплексе.

Эффективность оценивания может быть существенно повышена при наличии прототипов будущей программной системы. В этом случае эксперту предлагается оценить необходимость степени модернизации имеющегося прототипа.

Оценка трудозатрат, длительности и средней численности разработчиков при реализации проекта основывается на согласовании между разработчиком и заказчиком производительности труда программиста – (*P)* .

Среднестатистические оценки производительности труда программиста, полученные на основании многолетних наблюдений следующие [5]:
- при разработке программных систем первого класса сложности преимущественно на языке ассемблер – 60-80 строк/чел.-месяц;
- при разработке программных систем второго класса сложности (ИПС) на языках высокого уровня – 250-260 строк/чел.-месяц.

В таблице 3. представлены статистические показатели производительности (*P)* , рекомендуемые в базовой модели Constructive Cost Model (**COCOMO**) [5].

Таблица 3

Нормативы трудоемкости разработки программ (*P)*

Класс сложности ПС	Размеры ПС	
	до 30 тыс. строк	до 500 тыс. строк
Первый тип (**КПС**)	до 140 строк/чел.-месяц	до 80 строк/чел.-месяц
Второй тип (**ИПС**)	до 220 строк/чел.-месяц	до 160 строк/чел.-месяц
Третий тип (**ППП**)	до 250 строк/чел.-месяц	до 250 строк/чел.-месяц

Приведенные нормативы отражают не только трудоемкость непосредственного написания текстов программ, но и процессы комплексирования и испытания всего программного комплекса.

Прямой метод оценки технико-экономических показателей целесообразно использовать на ранних стадиях проектирования при разработке концепции и технического задания на будущую программную систему. Это позволит разработчику и заказчику определить трудоемкость реализации каждого функции, проранжировать их в соответствии с пожеланиями заказчика, соизмерить финансовые возможности заказчика и сроки реализации проекта.

1.2.2.2 Оценка трудозатрат методом функциональных точек

Оценка трудозатрат методом функциональных точек (Function point - **FP**) основывается на определении размера программной системы в терминах количества и сложности функций, реализуемых в данном программном коде [3].

Будущая система описывается в виде многоуровневой графической модели. Каждый из функций включает в себя входные и выходные данные, преобразования, внешние интерфейсы.

Процедура оценивания размеров программной системы состоит из следующей последовательности этапов:
- выделение множества функций (бизнес-процессов);
- оценка количества функциональных точек бизнес-процесса каждой категории;
- определение весовых коэффициентов сложности каждой функции;
- учет факторов и требований среды разработки программной системы;
- вычислений интегральных показателей сложности;
- вычисление итогового количества функциональных точек;
- определение размеров программного комплекса в показателях LOC (согласно табл. 1);
- определение размеров программной системы в целом.

При определении количества функций каждого бизнес-процесса следует руководствоваться следующими требованиями:
- учитываются только сложные функции, перечисленные в техническом задании (в соглашении о требованиях);
- при декомпозиции сложной функции учитываются все логические преобразования с данными.

Расчет количества функциональных точек по каждому бизнес-процессу сводиться в следующую таблицу (табл. 4).

В табл. 4 α_{ij} - весовой коэффициент сложности i -й функции j -й категории; x_{ij} - количество элементов данных i -й функции j -й категории.

Под **выводами** понимаются единицы информации, получаемые на выходе рассматриваемой функции:
- файлы, создаваемые в данной функции для передачи другим бизнес-процессам, либо за пределы программной системы;

- единицы деловой информации, предназначенные для конечных пользователей, оформленные в виде экранных форм, либо бумажных документов.

Таблица 4

Таблица функциональных точек

№	Категория функций	Простые	Средние	Сложные	Количество функциональных точек
1	Количество выводов	$\alpha_{11}x_{11}$	$\alpha_{12}x_{12}$	$\alpha_{13}x_{13}$	$X_1 = \sum_{j=1}^{3} \alpha_{1j}x_j$
2	Количество вводов	$\alpha_{21}x_{21}$	$\alpha_{22}x_{22}$	$\alpha_{23}x_{23}$	$X_2 = \sum_{j=1}^{3} \alpha_{2j}x_j$
3	Количество опросов вывода	$\alpha_{31}x_{31}$	$\alpha_{32}x_{32}$	$\alpha_{33}x_{33}$	$X_3 = \sum_{j=1}^{3} \alpha_{3j}x_j$
4	Количество опросов ввода	$\alpha_{41}x_{41}$	$\alpha_{42}x_{42}$	$\alpha_{43}x_{43}$	$X_4 = \sum_{j=1}^{3} \alpha_{4j}x_j$
5	Количество файлов	$\alpha_{51}x_{51}$	$\alpha_{52}x_{52}$	$\alpha_{53}x_{53}$	$X_5 = \sum_{j=1}^{3} \alpha_{5j}x_j$
6	Количество интерфейсов	$\alpha_{61}x_{61}$	$\alpha_{62}x_{62}$	$\alpha_{63}x_{631}$	$X_6 = \sum_{j=1}^{3} \alpha_{6j}x_j$
Общее количество функциональных точек					$F = \sum_{j=1}^{6} X_i$

Каждый из выводов, в зависимости от количества файлов, используемых при формировании выходов, следует отнести к одной из категорий сложности: простой, средний, сложный. В табл. 5 приводятся весовые коэффициенты сложности выводов.

Таблица 5

Весовые коэффициенты сложности вывода

Количество файлов	Количество элементов данных		
	1…5	6…19	≥20
1 (простой)	$\alpha_{11} = 4$	$\alpha_{12} = 4$	$\alpha_{13} = 5$
2÷3 (средний	$\alpha_{11} = 4$	$\alpha_{12} = 5$	$\alpha_{13} = 7$
≥4 (сложный)	$\alpha_{11} = 5$	$\alpha_{12} = 7$	$\alpha_{13} = 7$

Под **вводами** будем понимать единицы информации, поступающие на вход рассматриваемой функции:
- входные файлы, полученные из других бизнес-процессов, либо других программных систем;
- уникальная единица деловой информации, вводимая конечным пользователем.

По аналогии с выводом все вводы также рекомендуется разделять на простые, средние и сложные (см. табл. 6).

Под **опросами** понимаются действия, исполняемые программной системой в рассматриваемой функции:
- обращение к внешним процедурам, оформленным в виде команд или запросов, генерируемых извне и выполняемых программной системой;
- выполнение процедур, обеспечивающих непосредственный доступ к базе данных и выполняющих выборку с помощью простых ключей в режиме реального времени (но не выполняющих функции обновления).

Таблица 6

Весовые коэффициенты сложности ввода

Количество файлов	Количество элементов данных		
	1…5	6…19	≥20
1 (простой)	$\alpha_{21} = 4$	$\alpha_{22} = 4$	$\alpha_{23} = 5$
2÷3 (средний	$\alpha_{21} = 4$	$\alpha_{22} = 5$	$\alpha_{23} = 7$
≥4 (сложный)	$\alpha_{21} = 5$	$\alpha_{22} = 7$	$\alpha_{23} = 7$

Рекомендуется учитывать каждую уникальную единицу опроса, если:
- формат опроса отличается от формата ввода, вывода;
- формат опроса совпадает с форматом ввода, вывода, но требует дополнительной логики обработки.

При определении количества опросов не следует учитывать запросы к базам данных, использующие несколько ключей и выполняющие определенные операции, либо вычисления с последующим оформлением выводов.

Все опросы также рекомендуется разделять на простые, средние и сложные. В таблицах 7 и 8 приведены рекомендации по выбору весовых коэффициентов.

Таблица 7

Весовые коэффициенты сложности опросов вывода

Количество файлов	Количество элементов данных		
	1…5	6…19	≥20
1 (простой)	$\alpha_{31} = 4$	$\alpha_{32} = 4$	$\alpha_{33} = 5$
2÷3 (средний	$\alpha_{31} = 4$	$\alpha_{32} = 5$	$\alpha_{33} = 7$
≥4 (сложный)	$\alpha_{31} = 5$	$\alpha_{32} = 7$	$\alpha_{33} = 7$

Таблица 8

Весовые коэффициенты сложности опросов ввода

Количество файлов	Количество элементов данных		
	1…5	6…19	≥20
1 (простой)	$\alpha_{41} = 4$	$\alpha_{42} = 4$	$\alpha_{43} = 5$
2÷3 (средний	$\alpha_{41} = 4$	$\alpha_{42} = 5$	$\alpha_{43} = 7$
≥4 (сложный)	$\alpha_{41} = 5$	$\alpha_{42} = 7$	$\alpha_{43} = 7$

Под **файлами** будем понимать единицы информации, использующиеся программной системой в рассматриваемой функции:
- внутренние логические файлы программной системы;
- структуры данных, которые постоянно находятся внутри границ программной системы;
- внешние файлы, доступные пользователям с помощью ввода, вывода, опросов, либо интерфейсов.

Весовые коэффициенты оценки сложности файлов, в зависимости от количества взаимосвязей между таблицами представлены в табл. 9.

Под **интерфейсами**, используемыми рассматриваемой функцией, будем понимать:
- файлы, сгенерированных другими программными системами и использующихся в данной ПС;
- потоки данных, хранящихся за пределами программной системы, но используемых при управлении вычислительным процессом в любом направлении пересылки;
- структуры данных, использующихся в нескольких программных системах.

Весовые коэффициенты оценки сложности интерфейсов представлены в табл. 10.

Таблица 9

Весовые коэффициенты сложности структур данных (файлов)

Количество логических взаимосвязей	Количество элементов данных		
	1…19	20…50	≥51
1 (простой)	$\alpha_{51} = 5$	$\alpha_{52} = 5$	$\alpha_{53} = 7$
2÷5 (средний	$\alpha_{51} = 7$	$\alpha_{52} = 7$	$\alpha_{53} = 10$
≥6 (сложный)	$\alpha_{51} = 7$	$\alpha_{52} = 10$	$\alpha_{53} = 10$

Таблица 10

Весовые коэффициенты сложности интерфейсов

Количество логических взаимосвязей	Количество элементов данных		
	1…19	20…50	≥51
1 (простой)	$\alpha_{61} = 5$	$\alpha_{62} = 5$	$\alpha_{63} = 7$
2÷5 (средний	$\alpha_{61} = 7$	$\alpha_{62} = 7$	$\alpha_{63} = 10$
≥6 (сложный)	$\alpha_{61} = 7$	$\alpha_{62} = 10$	$\alpha_{63} = 10$

Общее количество функциональных точек определяется по следующей формуле:

$$F = \sum_{i=1}^{6} \sum_{j=1}^{3} \alpha_{ij} x_{ij}$$

Сложность предметной области и качества создаваемого программного обеспечения зависит от среды разработки приложений и требований конечных пользователей (**факторов и требований среды разработки**). Влияние этих факторов на размеры программного обеспечения оценивается по ряду показателей (табл.11). Каждый из показателей, в свою очередь, оценивается по пятибалльной шкале. Рекомендуемая шкала измерения показателей приведена в табл. 12.

Уровень влияния факторов внешней среды определятся по следующему соотношению [3]:

$$Z = 0{,}65 + 0{,}01 \times M \ , \qquad (3)$$

где M - суммарное значение весовых коэффициентов факторов внешней среды.

Уточненное количество функциональных точек с учетом факторов внешней среды определяется по следующей формуле [3]:

$$f = F \times Z \qquad (4)$$

Размерность программного обеспечения для конкретного языка программирования в **LOC** определяется с учетом нормативов, представленных в таблице 1 по соотношению:

$$R = f \times LOC \qquad (5)$$

Итоговая размерность программной системы определяется путем суммирования величины каждого R бизнес-процесса (функции).

Таблица 11

Факторы среды обработки

№	Фактор среды	Варианты
1	Каналы передачи данных	Входные и выходные данные передаются по локальной сети, магистральным каналам связи, по Internet
2	Распределенные вычисления	Пользовательские приложения используют данные, хранящиеся в едином хранилище, получают из других систем
3	Производительность системы	Скорость передачи данных и время отклика системы существенно для данной функции
4	Конфигурирование	Пользовательские приложения выполняются с применением интенсивно используемой, ограниченной, либо наполненной конфигурации
5	Частота транзакций	Использование приложений приводит к высокому сетевому трафику, экранные формы динамично изменяются, наблюдается высокая концентрация выходных форм, графика
6	Интерактивная обработка	Частота использования пользовательских приложений, участие пользователя при выполнении запросов
7	Пользовательский интерфейс	Значимость взаимодействия конечных пользователей с программной системой (необходимость дополнительного учета человеческого фактора)
8	Интерактивное обновление базы данных	Степень динамики обновления данных
9	Сложность обработки запросов	Уровень сложности алгоритмов обработки, количество транзакций, требования к безопасности и надежности
10	Сложность инсталляции программной системы	Наличие автоинсталляции, качество технической документации
11	Сложность эксплуатации системы	Наличие процедур запуска, резервирования, копирования, восстановления при ошибках, уровень (сложность) участия пользователей в этих процессах
12	Степень распределенности системы	Количество и удаленность пользовательских приложений
13	Гибкость изменения функций	Модульная реализация, наличие настроек, уровень поддержки со стороны пользователей, возможность изменения запросов

Таблица 12

Весовые коэффициенты факторов внешней среды

Влияние фактора не существенно	Влияние фактора существенно	Влияние фактора очень существенно
0÷1	2÷3	4÷5

1.2.2.3 Оценка длительности разработки ПС

1.2.2.3.1 Базовая модель оценки длительность разработки ПС

По оценке значения трудозатрат (R) длительность разработки системы (*T)* может быть определены следующим образом:

$$T = \frac{R}{P} \qquad (6)$$

где *P* - производительности труда программиста.

Если срок разработки (*D)* задан директивно заказчиком, исходя из реальных потребностей и наличия финансовых ресурсов, то средняя численность специалистов (*N)* , которые должны быть привлечены к реализации программной системы определяется по формуле:

$$N = \frac{T}{D} \qquad (7)$$

1.2.2.3.2 Оценка длительности разработки на основе базовой конструктивной модели - COCOMO

В основу оценки трудозатрат по базовой модели **COCOMO** положена степенная функция вида [3]:

$$T = \frac{A \times R^E}{12} \qquad (8)$$

где A - параметр прямо пропорциональный размерности программы R;
E - показатель степени, отражающий меру увеличении трудоемкости разработки каждой строки программного кода за счет увеличения количества взаимосвязей между компонентами;
R - размерность программной системы в тыс. строк (в тыс. **LOC**).

В таблице 13 приведены значения параметров A и E , полученные путем статистической обработки данных по результатам реализации различных проектов [5]. Оценки по **COCOMO** получены в результате обработки статистических данных по 160 реальным зарубежным проектам, а оценки по **ПРОМЕТЕЮ** — результат обобщения статистики – по 250 российским.

Таблица 13

Параметры модели оценки затрат

Тип **ПС**	**COCOMO**		**ПРОМЕТЕЙ**	
	A	E	A	E
Первый тип (**КПС**)	3,6	1,2	10	1,21
Второй тип (**ИПС**)	3	1,12	6,1	1,17
Третий тип (**ППП)**	2,4	1,05	-	-

Длительность разработки программной системы определяет общие сроки разработки **ПС**, начиная от разработки технического задания (требований) на систему и, завершая этапом проведения комплексных испытаний и может быть задана директивно заказчиком, исходя из реальных потребностей его бизнеса и наличия финансовых ресурсов. Средняя численность сотрудников**,** занятых в проекте, определяется по формуле (4).

1.2.2.3.3 Оценка длительности разработки ПС по распределению Релея

При разработке больших систем (свыше 50 чел.-лет) было установлено, что зависимость суммарных затрат от времени хорошо отображается следующим уравнением [7]:

$$E(t) = E_0(1 - e^{-at^2}),$$

где $E(t)$ - суммарные затраты к моменту времени t;
E_0 - общая стоимость системы;
a – характеристика максимальных затрат на единичном отрезке времени; (определяется исходя из состава исполнителей и производительности труда программистов).

Такая зависимость, выраженная в дифференциальной форме, отображается кривой Рэлея:

$$\dot{E}(t) = 2E_0 ate^{-at^2}$$

Максимум кривой $(\ddot{E}(T) = 0)$ соответствует длительности разработки системы. Таким образом, длительность разработки для больших систем может быть оценена по соотношению:

$$T = 12 \times \sqrt{\frac{R}{2 \times P \times N}}$$

где P - производительность труда программиста в чел.-лет (см. табл. 14);
N - численность участников разработки.

Таблица 14

Трудоемкость разработки программ (*P)*

№	Класс сложности **ПС**	*P* чел.-лет
1	Первый тип (**КПС**)	600
2	Второй тип (**ИПС**)	2000
3	Третий тип (**ППП**)	6000

1.2.2.4 Определение технико-экономических показателей проекта на основе размерности базы данных ПС

Размерность программной системы определяется количеством объектов, атрибутов и их взаимосвязями на объектных диаграммах бизнес-процессов.

Атрибут - простейший элемент базы данных информационной модели, содержащей одну из характеристик предметной области и вводимой либо непосредственно пользователем, либо заносящийся в базу из справочников и классификаторов.

Объект - элемент базы данных, формируемый из атрибутов и содержащий информацию о реальном процессе, явлении, предмете.

Размерность программного обеспечения определяется по следующей формуле:

$$R = 2 \times n \times 5 \times k \times 10 \times m, \qquad (9)$$

где n - количество объектов (таблиц) предметной области (количество связей между таблицами неограниченно и определяется структурой базы данных);

k - суммарное количество взаимосвязей между объектами;

m - количество атрибутов предметной области на один объект (количество связей между атрибутами определяется количеством источников формирования атрибутивной информации).

Таблица 15

Нормативы трудоемкости разработки программной системы

Категория сложности	θ, (чел./месяц)
Разработка прикладных программ (пользовательских приложений) с использованием стандартных средств СУБД: - количество прикладных программ (не более 3-х); - размерность базы данных (до 90 тыс. полей).	0,00467
Разработка прикладных программ (пользовательских приложений) с использованием стандартных пакетов прикладных программ: - количество прикладных программ (от 3-х до 10-ти); - размерность базы данных (от 90 тыс. до 200 тыс. полей).	0,00632
Разработка прикладных программ (пользовательских приложений) с использованием языков высокого уровня: - количество прикладных программ (не ограничено); - размерность базы данных (от 200 тыс. до 500 тыс. полей).	0,01233

Нормализованной величиной при создании программной системы является количество формируемых атрибутов, входящих в таблицы посредством установленных связей. При значениях n , k и m равных единице, величина, выражающая их количество равна 100. Длительность разработки определяются по формуле:

$$T = 0{,}01 \times R \times \theta, \qquad (10)$$

где θ - норматив трудоемкости разработки программной системы, и на основе статистических нормативов трудоемкости, приведенных в табл. 15.

Данный метод рекомендуется использовать при разработке программных систем на базе стандартных СУБД при:

- больших размерностях базы данных, формируемой из различных источников;
- наличии специализированных компонент, реализующих произвольные информационные запросы пользователей.

1.2.3 Оценка стоимости создания ПС

1.2.3.1 Расчет фонда оплаты труда

Расчет фонда оплаты труда основан последовательной реализации отдельных этапов жизненного цикла (**ЖЦ**) программного обеспечения с учетом:

- длительности каждого этапа жизненного цикла **ПС**;
- количественного и качественного состав специалистов, привлекаемых на каждом этапе жизненного цикла **ПС**;
- базовой месячной ставки специалиста-программиста.

В табл. 16, 17 приведены среднестатистические распределения первых двух величин по основным этапам жизненного цикла создания программных систем [7].

Таблица 16

Распределение трудозатрат и длительности по основным этапам жизненного цикла создания **ПС**

№	Этапы **ЖЦ**	Трудозатраты α (%)	Длительность β (%)
1	Анализ требований, предъявляемых к системе	10	10
2	Определение спецификаций	10	10
3	Проектирование	15	15
4	Кодирование	20	20
5	Тестирование (автономное и комплексное)	45	45

Средняя численность сотрудников, занятых на каждом из этапов создания программной системы определяется соотношением:

$$N_i = \frac{T \times \alpha_i}{D \times \beta_i} i = \overline{1{,}5} \qquad (12)$$

Таблица 17

Распределение специалистов по этапам **ЖЦ** программной системы

№	Этапы **ЖЦ**	Типы специалистов участвующих в проекте p_{ij} (%)		
		Аналитик	Программист	Технический специалист
1	Анализ требований, предъявляемых к системе	40	20	20
2	Определение спецификаций	60	20	20
3	Проектирование	35	35	30
4	Кодирование	10	65	25
5	Тестирование (автономное и комплексное)	15	60	25

Относительное распределение численности специалистов на каждом этапов жизненного цикла создания программной системы приведены в таблице 17.

Численность каждого типа специалистов на каждом из этапов жизненного цикла создания программной системы определяется по следующему выражению:

$$n_{ij} = \frac{p_{ij} \times N_i}{100}, i = \overline{1,5}, j = \overline{1,3} \qquad (13)$$

где p_{ij} - доля (%) специалистов j -го типа, привлекаемых для реализации проекта на i -ом этапе.

В том случае, фонд заработной платы для реализации i -го этапа проекта определяется как:

$$s_i = d_i \sum_{j=1}^{3} n_{ij} \times \upsilon_j, i = \overline{1,5} \qquad (14)$$

где d_i - длительность i -го этапа проекта;

υ_j - месячный фонд заработной платы j -го типа специалиста.

В основу определения υ_j может быть положена месячная базовая ставка программиста, размер которой может быть принят как одна из альтернатив:
- базовая ставка программиста заказчика;
- базовая ставка программиста разработчика;
- рыночная базовая ставка программиста в данном регионе (в качестве базовой ставки может быть принята стоимость разработки одной строки исходного кода программы 20-50 $ US [5]).

Соотношение месячной ставки специалиста-программиста к месячной ставке системного аналитика составляет как **1:1.3**, а к месячной ставке технического специалиста - как **1:0.7**.
Значения базовых окладов программистов ГОУ ВПО ТУСУР приведены в табл. 18.

Таблица 18

Базовые оклады программистов

№	Должность	Базовый оклад (грн./мес.)
1	Вед. инженер-программист	2200
2	Инженер-программист 1 категории	2000
3	Инженер-программист 2категории	1750
4	Программист	1660

1.2.3.2 Структура договорной цены

Структура договорной цены представлена в табл. 19.

Таблица 19

Структура договорной цены

№	Виды расходов	Всего на (грн.)
1	**2**	**3**
1	Заработная плата	Определяется по методике, изложенной в **разделе 1.2.3.1**
2	Прочие выплаты	Командировки и служебные разъезды в части оплаты суточных
3	Начисления на фонд оплаты труда (единый социальный налог) включая тариф на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний	% от п.п. 1-2 Определяется **Налоговым кодексом Украины**
4	Услуги связи	Телефоны, интернет – определяются **действующими расценками** на данные услуги

1	2	3
5	Транспортные услуги	В т.ч. оплата транспортных расходов при командировках и служебных разъездах. Определяется на **договорной основе**
6	Арендная плата за пользование имуществом	В т.ч. аренда помещений. Определяется на **договорной основе**
7	Услуги по содержанию имущества	Определяются на **договорной основе**
8	Прочие услуги, в т.ч. оплата проживания на время нахождения в служебной командировке	Определяется приказами **Минфина Украины** – для бюджетных организаций, на **договорной основе** – для организаций других форм собственности
9	Прочие расходы	Комплектация, расходные материалы и т.п. Определяются на **договорной основе**
10	Увеличение стоимости основных средств	Средства, связанные с производством ПС (здания, сооружения, оборудование, **вычислительная техника**, транспортные средства и др.)
Накладные расходы		**(15-20)% от п.п. 1-10.** Расходы на АУП, охрану, обслуживающий персонал и т. д.
Фонд развития производства		**(20-40)% от п.п. 1-10.** Устанавливается на **договорной основе между разработчиком и заказчиком**
ИТОГО		$\sum$ **по всем статьям сметы**

1.3 Оценка рыночной стоимости прикладного программного обеспечения

В тех случаях, когда разработка программного обеспечения ведется за счет собственных средств разрабатывающей организации необходимо проводить оценку рыночной стоимости разрабатываемой продукции и сроки окупаемости капиталовложений.

Результаты деятельности любой компании оцениваются величиной полученной прибыли. Для этого необходимо не только оценивать, сколько компания заработает при достижении запланированного объема продаж, но и определять, какой минимальный объем продаж необходим для обеспечения безубыточной работы. Этот минимально допустимый объем продаж, который покрывает все затраты на изготовление продукции, не принося при этом ни прибыли, ни убытков, получил название *точка безубыточности* (break-event point).

Точка безубыточности - это такой объем продаж продукции фирмы, при котором выручка от продаж полностью покрывает все расходы на производство продукции. (В том числе среднерыночный процент на собственный капитал фирмы и нормальный предпринимательский доход).

Для максимизации прибыли необходимо соблюдать соотношение между предельным доходом и предельными издержками при увеличении выпуска на одну единицу. В случае, когда предельные издержки меньше предельного дохода, увеличение выпуска повлечет за собой увеличение дохода фирмы.

1.3.1 Виды и составляющие издержек

Все затраты разделяют на две группы: *условно-переменные* и *условно-постоянные.*

FC (Fixed Cost) – *постоянные издержки* – денежные издержки, в целом не изменяющиеся в зависимости от изменения объёма выпускаемой продукции. К основным составляющим фиксированных издержек можно отнести следующие виды затрат:

- зарплата руководящего состава и административного персонала (служащих);
- аренда и лизинг;
- амортизация зданий и оборудования;
- налоги (подоходные, на платежную ведомость, и т.д.);
- платежи за внешние услуги;
- плата за телефон и коммунальные услуги;
- плата за страхование;

- уплата процентов по кредитам;
- общие административные расходы.

VC (Variable Cost) – *переменные издержки* – издержки, меняющиеся пропорционально объёму производства. Переменными издержками являются:
- затраты на сырьё и труд основных производственных рабочих (заработная плата);
- комиссионные отчисления торговым агентам;
- затраты на приобретение тары, упаковочного материала;
- транспортные расходы.

Деление затрат на постоянные и переменные даёт возможность:
- определить сроки окупаемости затрат;
- определить запас финансовой прочности предприятия;
- рассчитать оптимальную величину прибыли предприятия.

Кроме этих затрат выделяют *смешанные издержки* (постоянно-переменные затраты), которые включают в себя элементы как постоянных, так и переменных расходов: оплата топлива, почтовые расходы, телефон, отопление, затраты на текущий ремонт оборудования, электроэнергию и т. д. Постоянно-переменные затраты увеличиваются при увеличении объёма производства, но не пропорционально его росту, а в другой пропорции. При конкретных расчётах необходимо выделять в составе смешанных издержек постоянную и переменную части, причисляя их к соответствующему виду затрат.

Текущие затраты, обеспечивающие жизнедеятельность предприятия – это постоянная составляющая постоянно-переменных затрат, а затраты, связанные с развитием производства, – это переменная составляющая.

1.3.2 Определение точки безубыточности

Основным методом определения точки безубыточности является CVP-анализ (Cast Value Profit - затраты, объем, прибыль), основанный на анализе соотношений затрат и объемов выпуска и прибыли.

Чистая прибыль фирмы определяется как разница между выручкой и переменными и постоянными издержками.

$$P = C \times \nu - \left(F_c + V_c \times \nu\right) = \left(C - V_c\right) \times \nu - F_c, \qquad (15)$$

где P - прибыль фирмы;

ν - объем выпуска продукции;

C - договорная цена продажи единицы продукции;

F_c - величина фиксированных расходов (FC);

V_c - величина переменных издержек на единицу продукции (VC).

В этом случае объем выпуска, при котором достигается точка безубыточности, определяется как:

$$\nu_0 = \frac{F_c}{C - V_c} \qquad (16)$$

В случае если объем продаж определен (ν_0 заранее известен) рыночная цена единицы продукции при нулевом уровне прибыли определяется как:

$$C_0 = \frac{F_c + V_c \times \nu_0}{\nu_0} \qquad (17)$$

Если фирма хочет получить дополнительную прибыль, и рыночная цена C_0 известна, то объем продаж при заданном уровне прибыли P_0 можно определить по следующей формуле:

$$\nu_0 = \frac{P_0 + F_c}{C_0 - V_c} \qquad (18)$$

Корректное использование описанного подхода возможно лишь для ограниченного объема выпуска. Это объясняется тем, что при большом объеме выпуска продукции перестают быть верными предпосылки, лежащие в основе CVP-анализа, например, неизменный характер и величина постоянных расходов, и т.д. Выделяют следующие условия применимости CVP-анализа при определении точки безубыточности:

- цены на продукцию, материалы и услуги, используемые в производстве, неизменны;
- производительность труда специалистов постоянна;
- ассортимент продукции и объемы выпуска каждого вида неизменны - отсутствуют структурные сдвиги в производстве;
- на конец анализируемого периода объем производства равен объему продаж, либо изменения начального и конечного уровня запасов незначительны.

1.4. Оценка эффективности проекта

Современный рынок программного обеспечения предлагает довольно широкий набор программных комплексов и корпоративных информационных систем. Однако использование таких систем не всегда является целесообразным, так как возникает необходимость в закупке и сопровождении системы, а это нерационально, если *на предприятии уже используется* какая-нибудь другая система. В этих случаях иногда обоснованным является разработка программной системы *улучшающей* (*или/и расширяющей*) функции уже существующей.

Для оценки конкурентоспособности разрабатываемого продукта в этом случае необходимо провести анализ и сравнение аналогом по функциональному назначению, основным техническим и эксплуатационным параметрам, областям применения. Подобный анализ осуществляется с помощью оценки эксплуатационно-технического уровня разрабатываемого продукта. Такой анализ подразумевает оценку *коэффициента технического уровня* программного продукта, годового *экономического эффекта* от использования разрабатываемой системы, *коэффициент экономической эффективности* разработки и *срока окупаемости* затрат на разработку продукта.

1.4.1 Оценка конкурентоспособности ПС в сравнении с аналогом

Эксплуатационно-технический уровень (ЭТУ) разрабатываемого продукта – это обобщенная характеристика его эксплуатационных свойств, возможностей, степени новизны, являющихся основой качества продукта. Для определения ЭТУ продукта можно использовать индекс эксплуатационно-технического уровня J , который рассчитывается как сумма частных индексов, куда входят показатели качества программного продукта.

Таблица 20.

Показатели качества

Показатели качества	Весовой коэффициент B_i	Проект		Аналог	
		X_i^1	$B_i \times X_i^1$	X_i^2	$B_i \times X_i^2$
1. Удобство работы (пользовательский интерфейс)	0,1				
2. Новизна (соответствие современным требованиям)	0,06				
3. Соответствие профилю деятельности заказчика	0,15				
4. Операционная система (многозадачность, графика)	0,05				
5. Надежность (защита данных)	0,13				
6. Скорость доступа к данным	0,09				
7. Гибкость	0,05				
8. Функции обработки информации	0,13				
9. Соотношение стоимость/возможности	0,09				
10. Время обучения персонала	0,15				
Обобщенный показатель качества J		$J_1 = \sum_{i=1}^{10} B_i \times X_i^1$		$J_2 = \sum_{i=1}^{10} B_i \times X_i^2$	

$$J = \sum_{i=1}^{n} B_i \times X_i \,, \qquad (19)$$

где n – число рассматриваемых показателей;

B_i – коэффициент весомости i -го показателя в долях единицы;

X_i – относительный показатель качества, устанавливаемый экспертным путем по выбранной шкале оценивания $\left(X_i = \overline{1,5}\right)$.

Для учета значимости отдельных параметров применяется балльно-индексный метод. Результаты расчета балльно-индексным методом по 5-ти балльной шкале оценивания оформляются в виде таблицы (см. табл. 20).

Отношение двух найденных индексов называют коэффициентом технического уровня A проектируемого программного продукта по отношению к аналогу.

$$A = \frac{J_1}{J_2} \qquad (20)$$

Разработка системы технологически обоснованна если значение $A > 1, 0$.

1.4.2 Расчет показателя экономического эффекта

Оценка экономической эффективности проекта основывается на расчете показателей сравнительной экономической эффективности капитальных вложений. Экономический эффект проекта E определяется по разности приведенных затрат базового и нового варианты ПС:

$$E = W_2 \times A - W_1, \qquad (21)$$

где W_1, W_2 - приведенные затраты на единицу работ, выполняемых с помощью проектируемого и базового (аналога) вариантов, грн.;

A - коэффициент технического уровня, определяемый по формуле (20).

Приведенные затраты W_i на единицу работ, выполняемых по базовому и разрабатываемому вариантам, рассчитываются по формуле

$$W_i = C_i + E_n \times K_i; i = \overline{1,2} \qquad (22)$$

где C_i - себестоимость (эксплуатационные затраты), грн.;

E_n - нормативный коэффициент экономической эффективности $\left(E_n = 0{,}33\right)$;

K_i - суммарные затраты, связанные с разработкой проекта (аналога). (Смотри табл. 19).

1.4.3 Расчет эксплуатационных затрат

К эксплуатационным затратам относятся затраты, связанные с обеспечением нормального функционирования проекта. Это могут быть затраты на ведение информационной базы, эксплуатацию комплекса технических средств, эксплуатацию систем программно-математического обеспечения, реализацию технологического процесса обработки информации по задачам, эксплуатация системы в целом.

Эксплуатационные затраты включают в себя:

- затраты на зарплату основную и дополнительную с отчислениями во вне-бюджетные фонды;
- амортизационные отчисления от стоимости оборудования и устройств системы;
- затраты энергию;
- затраты на текущий ремонт оборудования и устройств системы;
- затраты на материалы и машинные носители;
- накладные расходы.

Расчет эксплуатационных затрат (*себестоимости*) *проводится по методике изложенной в разделе* 1.2.3 *и приводится в виде сметы затрат*. Базовый расчетный срок 1 год.

1.4.4 Расчет срока окупаемости затрат

Срок окупаемости затрат на разработку ПС оценивается по формуле:

$$T_0 = \frac{K_1}{E} \qquad (23)$$

Фактический коэффициент экономической эффективности разработки E_f определяется как величина обратная сроку окупаемости.

$$E_f = \frac{1}{T_0} \qquad (24)$$

Если фактический коэффициент экономической эффективности разработки E_f получился больше, чем нормативный E_n, то разработка и внедрение разрабатываемой ПС является экономически обоснованной.

2. Практическая часть

Практическая часть заключается в выполнении студентами пяти лабораторных работ, которые охватывают весь комплекс технико-экономического обоснования выполнения программных проектов:

1. Лабораторная работа №1 “Описание программного продукта”.
2. Лабораторная работа №2 “Определение технико-экономических показателей проекта”.
3. Лабораторная работа №3 “Определение договорной цены на создание программной системы”.
4. Лабораторная работа №4 “Расчет точки безубыточности”.
5. Лабораторная работа №5 “Расчет технико-экономической эффективности разработки программной системы”.

Лабораторные работы выполняются студентами в рамках примерных тем программных проектов (студент может предложить свою тему программного проекта):

1. Автоматизированная система учета клиентов организации.
2. Программное обеспечение интернет-магазина по продаже стройматериалов.
3. Автоматизированная система контроля и обращения клиентов в службу технической поддержки «Техцентр».
4. Автоматизированная система информационного справочника ОАО.
5. Автоматизированная система расчета производственной мощности предприятия.
6. Программное обеспечение автоматизированного рабочего места инспектора финансового отдела райгосадминистрации.
7. Автоматизированная система подразделения высшего учебного заведения «Деканат».
8. Автоматизированная информационная система института на основе WEB-интерфейса.
9. Автоматизированная система управления малым предприятием.
10. Автоматизированная система подразделения высшего учебного заведения «Факультет».

Зачет каждой лабораторной работы проводится в индивидуальном порядке в форме собеседования студента с преподавателем. Отчет по лабораторной работе оформляется в электронном или бумажном виде.

Защита экономической части программного проекта проводится в форме устного доклада (презентация – обязательна) и обсуждения на семинаре.

Приложение

Пример оформления технико-экономического обоснования проекта

.

1. Описания программного продукта

«Автоматизированная информационная система «Торговля и склад», версия 1.1.(АИС «ТиС»). Поставщик – малая фирма “Тамир”. АИС «ТиС» предназначена для информационной поддержки процессов управления складами и торговыми представительствами заказчика.

АИС «ТиС» соответствует ГОСТ ИСО/МЭК 12119-2000 «Информационная технология. Пакеты программ. Требование к качеству и тестирование».

Программно-технические средства.

Программное обеспечение АИС «ТиС» разработано в архитектуре «клиент-сервер», средой разработки является язык программирования Delphi, СУБД – Firebird 1.5, под управлением операционной системы MS Windows 2000/XP;

Техническое обеспечение:

Сервер:
- CPU Intel Pentium IV 2,6 GHz/1024 Mb,
- HDD 160Gb SATA,
- сеть 1000/100 Mb/s,
- ОС MS Windows 2000/XP,
- СУБД – Firebird 1.5;

Клиент:
- CPU Intel Celeron 1,3 GHz/128 Mb,
- HDD 120 Gb,
- ОС MS Windows 2000/ XP,
- СУБД – Firebird 1.5.

В комплект поставки входят:

1) оптический носитель (компакт-диск), содержащий дистрибутив системы в загрузочных модулях и инсталляционные пакеты, в том числе:
- клиентская часть, представляющая собой программное обеспечение для работы с базой данных АИС;
- серверная часть, представляющая собой файл данных БД АИС «ТиС»;
- инсталляционные пакеты СУБД Firebird 1.5.1. под Linux и Windows, являющейся клоном Interbase и разрабатывающейся как продукт Open Source.

2) техническая документация в составе:
- АИС «ТиС». Техническое задание.
- АИС «ТиС». Общее описание системы.
- АИС «ТиС». Руководство пользователя.
- АИС «ТиС». Руководство программиста.

Документация выполнена в соответствии с ГОСТ ИСО/МЭК 15910—2002. «Информационная технология. Процесс создания документации пользователя программного средства».

Инсталляция системы.

АИС «ТиС» снабжена программой инсталляции, обеспечивающей установку и настройку программного обеспечения в локальной сети (сервер) и на отдельные компьютеры (рабочие станции), а также инструкцию по установке инсталляционных пакетов СУБД Firebird 1.5.1..

Функциональные возможности системы.

АИС «ТиС» реализована как распределенная система управления процессами закупки и реализации продукции в составе комплекса взаимосвязанных АРМ («Администратор», «Прием продукции», «Прием заказов», «Прием платежей», «Исполнение заказов»), объединенных единым форматом представления данных, идеологией обработки информации и ориентированных на использование баз данных общего пользования, единую техническую базу и операционную среду.

Для оптимизации процессов сбора и обработки информации, минимизации затрат на ввод и хранение данных, повышения актуальности, достоверности и сопоставимости данных различных информационных систем в АИС «ТиС» реализованы следующие технологические требования:
- централизованная база данных, с возможностью подключения к ней с удаленных терминалов посредством сети Интернет;
- однократный ввод данных в систему с возможностью дальнейшего их использования в функционально связанных подсистемах;
- включение в состав хранимой БД информации только тех данных, для которых существуют надежные тракты актуализации;
- своевременная актуализация данных в базе в зависимости от вида хранимой информации.

АИС «ТиС» выполняет следующие функции:
- внесение, поиск и обработка информации о поставщиках и поставляемых продуктах;
- внесение, поиск и обработка информации о заказчиках и заказах;
- внесение, поиск и обработка информации о платежах;
- проведение товарно-финансовых операций при исполнении заказов;
- администрирование АИС, в том числе ведение словарей, справочников, классификаторов, списков пользователей.
- разграничение доступа к данным и функциям АИС на основе системы ролей и привилегий.

Надежность. АИС «ТиС» обеспечивает круглосуточную бесперебойную работу при отсутствии помех со стороны аппаратного обеспечения и операционной системы.

Предусмотрены механизмы автоматического резервного копирования данных, возможна настройка периодичности выполнения резервного копирования.

Предусмотрена возможность восстановления системы по команде администратора на основе последних результатов резервного копирования.

В случае возникновения системного сбоя (отключение питания, неполадки в аппаратном обеспечении, в работе операционной системы) АИС «ТиС» выполняет самостоятельное восстановление работоспособности после перезагрузки системы.

Практичность.

Взаимодействие с пользователем организовано посредством графического пользовательского интерфейса в общепринятой форме. Предусмотрены возможности настройки пользовательского интерфейса под требования пользователя: возможна настройка системы меню, внешнего вида приложения. Настройки хранятся на сервере БД и действуют для конкретного пользователя из любой точки доступа к АИС.

Для облегчения работы пользователя в АИС «ТиС» предусмотрена контекстная справочная система, содержащая информацию о порядке работы с конкретными функциями АИС.

Эффективность использования системы заключается в:
- сокращении времени выполнения расчетов по текущим торговым операциям;
- сокращении времени построения оперативных и регламентных отчетов;
- повышении качества и контроля исполнительной дисциплины сотрудников организации.

Время отклика системы на запрос пользователя при условии, что аппаратная конфигурация удовлетворяет рекомендуемым системным требованиям, не превышает:
- 1 сек. – при выполнении оперативных внесений в БД;
- 10 сек. – при построении оперативных отчетов;
- 5 мин. – при построении регламентных отчетов.

Построение регламентных отчетов происходит в фоновом режиме (пользователь может продолжать работу с системой в процессе построения регламентного отчета).

Время восстановления работоспособности системы после перезагрузки не превышает 15 минут, при условии, что аппаратная конфигурация удовлетворяет рекомендуемым системным требованиям.

Сопровождаемость.

В АИС «ТиС» предусмотрена возможность уведомления администратора по электронной почте о возникновении сбоев в работе системы. Специальные возможности изменения режимов функционирования системы не предусмотрены. Специальные возможности модификации системы не предусмотрены.

Мобильность.

Возможность перенесения АИС «ТиС» на другую аппаратно-программную платформу не предусмотрена.

Надежность, практичность, эффективность, сопровождаемость и мобильность системы соответствует основным положениям ГОСТ 28195-89 “Оценка качества программных средств. Общие положения”.

2. Определения технико-экономических показателей проекта

Исходные данные:

- Тип системы: информационно-справочная.
- Сложность системы: простая.
- Язык программирования – Delphi.
- Плановый срок разработки системы – 12 месяцев.

В результате анализа системы получаем следующую структуру программных комплексов и компонент (рис. 1).

Размеры программной системы определяется в виде количества строк исходного кода в терминах Lines of code-**LOC** (см. табл. 1, колонка 3). В качестве показателя количества строк исходного кода используется число операторов языка ассемблер. Так как при разработке системы используется язык Delphi., то этот показатель LOC на 1 функциональную точку равен **29**.

(***Вариант 1 – Прямой метод оценки трудозатрат***)

Оценку проводит один эксперт. Результаты экспертизы приведены в таблице 21.

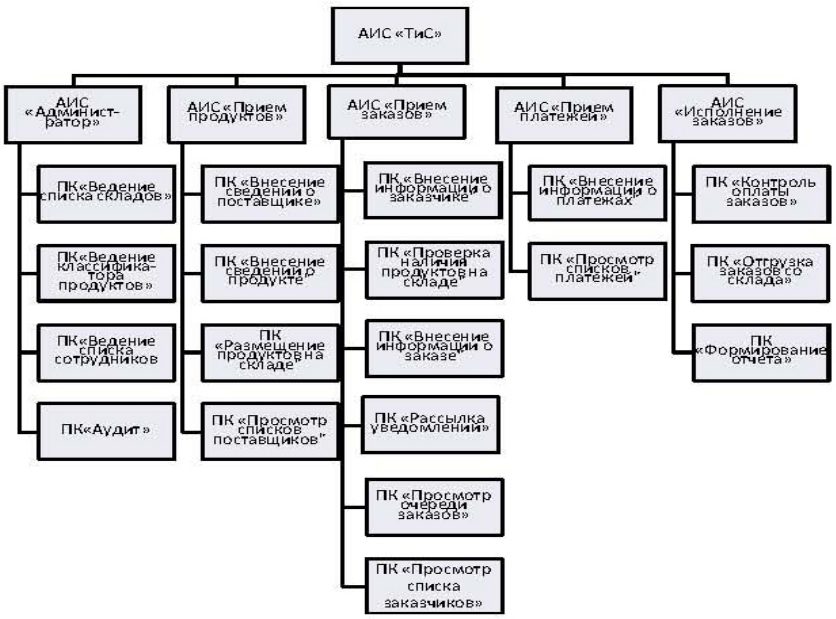

Рис. 1 Декомпозиция АИС «ТиС»

Оценка трудозатрат, длительности и средней численности разработчиков при реализации проекта основывается на согласовании между разработчиком и заказчиком производительности труда программиста - P . Согласно нормативам трудоемкости разработки программ в базовой модели COCOMO (табл. 3) примем P = 220 строк/чел.-месяц (простая информационно-справочная система, количество строк – до 30 тыс.). Трудозатраты на разработку определяются по формуле (6):

$$T = \frac{R}{P} = \frac{4372}{220} = 19{,}8727 \text{ (чел.- месяцев).}$$

При заданной длительности разработки (12 месяцев) среднюю численность персонала определяем по формуле (7):

$$N = \frac{T}{D} = \frac{19{,}8727}{12} = 1{,}656 \text{ (чел.)}.$$

Таким образом, основные технико-экономические показатели разработки:
- трудозатраты на разработку системы за 12 месяцев составят 19,872 человеко-месяцев;
- необходимые людские ресурсы при реализации системы за 12 месяцев - 1,656 чел.

Таблица 21

Экспертная оценка размерности программной системы

Программные комплексы и компоненты АИС «ТиС»	Пес.	Реал.	Опт.	Средняя оценка (бета-распр.)
АИС "Администратор"				
ПК «Ведение списка складов»	96	64	32	64
ПК «Ведение классификатора продуктов»	32	32	32	32
ПК «Ведение списка сотрудников»	32	32	32	32
ПК «Аудит»	320	160	120	180
АИС «Прием продуктов»				
ПК «Внесение сведений о поставщике»	640	320	128	341,33
ПК «Внесение сведений о продукте»	640	320	128	341,33
ПК «Размещение продуктов на складе»	128	128	128	128
ПК «Просмотр списков поставщиков»	128	128	128	128
АИС «Прием заказов»				
ПК «Внесение информации о заказчике»	640	320	128	341,33
ПК «Проверка наличия продуктов на складе»	640	320	128	341,33
ПК «Внесение информации о заказе»	640	320	128	341,33
ПК «Рассылка уведомлений»	320	320	128	288
ПК «Просмотр очереди заказов»	32	320	128	240
ПК «Просмотр списка заказчиков»	32	320	128	240
АИС «Прием платежей»				
ПК «Внесение информации о платежах»	512	320	64	309,33
ПК «Просмотр списков платежей»	640	512	64	458,66
АИС «Исполнение заказов»				
ПК «Контроль оплаты заказов»	128	64	64	74,66
ПК «Отгрузка заказов со склада»	128	64	64	74,66
ПК «Формирование отчета»	320	512	128	416
ИТОГО				4372

(***Вариант 2*** – метод функциональных точек)

Размеры программной системы оцениваются в терминах количества и сложности функций (бизнес-процессов), реализуемых в данном программном коде. Система описывается в виде многоуровневой графической модели, представленной в виде совокупности пользовательских бизнес-процессов (рис. 1). Каждая из функций включает в себя входные и выходные данные, преобразования, внешние интерфейсы. Процедура оценивания размеров программной системы из следующей последовательности этапов: *ввод,вывод, опросы, структуры данных, интерфейсы.*

На основании методики, изложенной в разделе 1.2.2.2. проводятся расчет количество функциональных точек по каждой функции (бизнес-процессу) и заполняются соответствующие таблицы по аналогии с таблицей 4. При этом используем весовые коэффициенты сложности выводов, вводов, опросов ввода, опросов вывода, сложности структурных данных (файлов), сложности интерфейсов (табл. 5 – 10).

Для разрабатываемой АИС «ТиС» получаем рабочие таблицы определения количества функциональных точек по каждому бизнес-процессу (табл. 22).

Общее количество функциональных точек по всем бизнес процессам составит:

$$F = 304 + 298 + 297 + 296 + 276 = 1471.$$

Размерность программной системы определяется с учетом факторов и требований среды разработки (конечных пользователей системы)**,** так как от этих факторов зависит сложность предметной области и качество создаваемого программного обеспечения.

Таблица 22

Рабочая таблица определения количества функциональных точек

Категория функций	Простые	Средние	Сложные	Кол-во точек
АИС «Администратор»				
Количество выводов	0	5×8	0	40
Количество вводов	0	5×8	0	40
Количество опросов вывода	0	0	8×7	56
Количество опросов ввода	0	0	8×6	48
Количество файлов	0	10×3	0	30
Количество интерфейсов	0	0	9×10	90
Количество функциональных точек				**304**
АИС «Прием продуктов»				
Количество выводов	0	5×4	7×4	48
Количество вводов	4×1	5×1	0	9
Количество опросов вывода	4×1	0	0	4
Количество опросов ввода	0	0	10×6	60
Количество файлов	7×1	10×4	0	47
Количество интерфейсов	0	0	10×13	130
Количество функциональных точек				**298**
АИС «Прием заказов»				
Количество выводов	0	5×2	7×2	24
Количество вводов	4×1	5×1	7×1	16
Количество опросов вывода	0	0	7×1	7
Количество опросов ввода	0	0	6×10	60
Количество файлов	0	15×4	0	60
Количество интерфейсов	0	0	10×13	130
Количество функциональных точек				**297**
АИС «Прием платежей»				
Количество выводов	0	5×4	0	20
Количество вводов	4×1	5×1	0	9
Количество опросов вывода	0	0	7×1	7
Количество опросов ввода	0	0	15×6	90
Количество файлов	0	10×5	0	50
Количество интерфейсов	0	0	10×12	120
Количество функциональных точек				**296**
АИС «Исполнение заказов»				
Количество выводов	0	4×5	0	20
Количество вводов	0	0	8×6	48
Количество опросов вывода	0	0	7×1	7
Количество опросов ввода	0	0	7×3	21
Количество файлов	0	10×8	0	80
Количество интерфейсов	0	0	10×10	100
Количество функциональных точек				**276**

Влияние этих факторов на размеры программного обеспечения оценивается по показателям, приведенным в таблице 11. При этом каждый из показателей, в свою очередь, оценивается по пятибалльной шкале измерения, которая приведена в табл. 12. (оценка

существенности влияния факторов среды). Учитывая вышеизложенное, проводится оценка влияния данных факторов (таблица 23).

Таблица 23.

Факторы и требования среды разработки

№	Факторы среды	Значение
1	Каналы передачи данных	4
2	Распределенные вычисления	1
3	Производительность системы	5
4	Конфигурирование	2
5	Частота транзакций	2
6	Интерактивная разработка	2
7	Пользовательский интерфейс	5
8	Интерактивное обновление БД	3
9	Сложность обработки запросов	4
10	Сложность установки ПО	5
11	Сложность эксплуатации системы	5
12	Степень распределенности системы	3
13	Гибкость изменения функций	4
Суммарное значение коэффициентов (*M*)		**45**

Влияние факторов внешней среды оценивается по формуле (3):

$$Z = 0,65 + 0,01 \times M = 0,65 + 0,01 \times 45 = 1,1.$$

Уточненное количество функциональных точек, с учетом факторов внешней среды определим по формуле (4):

$$f = F \times Z = 1471 \times 1,1 = 1618.$$

Размерность программного обеспечения для конкретного языка программирования определим с учетом нормативов, представленных в таблице 2 по формуле (5):

$$R = f \times \mathbf{LOC} = 1618 \times 29 = 46922.$$

Оценка трудозатрат проводится с помощью степенной функции вида (8).

Значения параметров A и E определяются из таблицы коэффициентов математической модели оценки трудозатрат в зависимости от типа программной системы (табл. 13) $A = 3$, $E = 1,12$.

$$T = \frac{A \times R^E}{12} = \frac{3 \times 46,922^{1,12}}{12} = 18,619 \text{ (чел.– месяцев)}$$

Средняя численность сотрудников, занятых в проекте, определяется по формуле (7) и составляет

$$N = \frac{T}{D} = \frac{18,616}{12} = 1,551$$

Таким образом, метод функциональных точек определил следующие основные технико-экономические показатели:

- трудозатраты на разработку системы за 12 месяцев составят 18,616 человеко - месяцев;
- необходимые людские ресурсы при реализации системы за 12 месяцев - 1,551 чел.

(***Вариант 3*** – метод на основе размерности базы данных)

В результате анализа объекта с помощью *ER*-моделирования строим концептуальную модель базы данных программной системы для определения количества таблиц (объектов) предметной области, связей и атрибутов.

Анализируя построенную модель БД получаем:

- количество таблиц (объектов) $n = 9$
- количество взаимосвязей между объектами $k = 12$
- количество атрибутов на один объект $m = 23$

Размерность программного обеспечения (базы данных) определяется по формуле (9):

$$R = 2 \times n \times 5 \times k \times 10 \times m = 2 \times 9 \times 5 \times 12 \times 10 \times 23 = 248400 \text{ полей БД}$$

Трудозатраты определяются по формуле (10) на основе статистических нормативов трудоемкости, приведенных в табл. 15.

Для размерности базы данных 194400 полей норматив трудоемкости находится в нормативном промежутке от 90 тыс. до 200 тыс. полей, что соответствует значению норматива $\theta = 0{,}00632$. Таким образом, трудоемкость будет равна:

$$T = 0{,}01 \times R \times \theta = 0{,}01 \times 248528 \times 0{,}00632 = 15{,}7 \text{ (чел.-месяцев).}$$

Длительность разработки12 месяцев, тогда средняя численность специалистов, которые должны быть привлечены к реализации ПС, определяется по известной формуле:

$$N = \frac{T}{D} = \frac{15{,}7}{12} = 1{,}309 \text{ (чел.)}$$

Таким образом, применяя метод определения ТЭП на основе размерности базы данных ПС, определили следующие основные технико-экономические показатели разработки:
- трудозатраты на разработку системы за 12 месяцев составят 15,707 человеко-месяцев;
- необходимые людские ресурсы при реализации системы за 12 месяцев - 1,309 чел.

3 Определение договорной цены на создание программной системы

3.1 Определение фонда оплаты труда на разработку и комплексные испытания программной системы

В основу определения фонда оплаты труда положены:
- длительность реализации каждого этапа жизненного цикла проекта;
- количество и качественный состав специалистов, привлекаемых на каждом этапе проекта;
- базовая месячная ставка специалиста-программиста.

Выбираем исходные данные, полученные с помощью метода определения ТЭП на основе размерности БД:
трудоемкость - 15,7 чел.-месяцев
длительность - 12 месяцев
средняя численность специалистов - 1,309 человек

Формируем таблицу средней численности сотрудников, занятых на каждом из этапов создания ПС, используя статистические данные таблицы 16 и по формуле (12) выполняем расчет средней численности сотрудников, занятых на каждом из этапов создания ПС.

Таблица 24

Средняя численность сотрудников, занятых на каждом из этапов создания программной системы и длительности каждого этапа

№	Этапы **ЖЦ**	Численность сотрудников, чел	Длительность (месяцев)
1	Анализ требований, предъявляемых к системе	1,309	1,2
2	Определение спецификаций	1,309	1,2
3	Проектирование	1,309	1,8
4	Кодирование	1,309	2,4
5	Тестирование (автономное и комплексное)	1,309	5,4

Распределение специалистов по этапам жизненного цикла программной системы оценивается путем расчетов по соотношению (13) с использованием статистического распределения таблицы 17.

Фонд заработной платы для реализации i -го этапа проекта определим по формуле (14). Примем размер ставки программиста равной 4 тысячи гривень, как рыночную базовую ставку программиста в данном регионе. Тогда в соответствии с принятыми соотношениями ставки участников разработки будут: базовая ставка программиста 3 750 грн., ставка аналитика 4875 грн., ставка техника 2625 грн. Результаты расчета фонда зарплаты сведены в таблицу 26.

Таблица 25

Численность каждого типа специалистов на каждом из этапов жизненного цикла создания программной системы

№	Этапы **ЖЦ**	Типы специалистов, чел. (n_{ij})		
		Аналитик	Программист	Технический специалист
1	Анализ требований, предъявляемых к системе	0,52360	0,26180	0,52360
2	Определение спецификаций	0,78540	0,26180	0,26180
3	Проектирование	0,45815	0,45815	0,39270
4	Кодирование	0,13090	0,85085	0,32725
5	Тестирование (автономное и комплексное)	0,19635	0,78540	0,32725

Таблица 26

Распределение фонда заработной платы по этапам жизненного цикла программной системы

Этапы **ЖЦ**	Аналитик	Программист	Технический специалист	ФЗП по этапу
Анализ требований, предъявляемых к системе	3063,06	1178,1	1649,34	5891,1
Определение спецификаций	4594,59	1178,1	824,67	6597,36
Проектирование	4020,27	3092,51	1855,51	8968,29
Кодирование	1531,53	7657,65	2061,68	11250,86
Тестирование (автономное и комплексное)	5168,92	15904,35	4638,77	23134,95
Итого общий фонд заработной платы				55842,56

3.2 Структура договорной цены на программное обеспечение

Основополагающим элементом, из которого производиться расчет стоимости проекта, является рассчитанный выше общий фонд заработной платы.

Разделы сметы затрат зависят от формы организации разработчика (государственное предприятие, коммерческое) и соответствующих форм налогообложения ее деятельности. В приведенном ниже примере предполагается, что система разработана в коммерческой организации, реализующей продукцию и услуги с обычной системой налогообложения, предусматривающей налог на добавленную стоимость (определяется законодательством Украины, для расчетов принимаем 18%).

Необходимые виды основных расходов, из которых и складывается окончательная смета затрат (командировки, коммунальные услуги, прочие расходы, накладные расходы и т.д.) определяются исходя из производственного процесса.

Например, предположив, что стоимость приобретенной для выполнения проекта компьютерной техники (программно-аппаратный комплекс) составит 15 тыс. гривень, мы должны иметь в виду, что амортизационные отчисления для средств вычислительной техники, согласно действующему законодательству будут производиться в течение пяти лет, т.е. составят 15000/5 = 3000 гривень в год. (те же 12 месяцев, планируемых на разработку системы).

Условно приведены командировочные расходы. Процент накладных расходов также не имеет жестких нормативов и зависит от затрат на содержание АУП, бухгалтерии и т.д.

Итак, с учетом нормативов, приведенных в таблице 19. смета затрат и общая стоимость проекта сведены в таблицу 27. Договорная цена на разработку и внедрение программной системы АИС «ТиС» составляет: 140000 гривень.

Таблица 27.

Смета затрат на разработку и внедрение системы

Наименование статей расходов	**Сумма (грн.)**
Фонд оплаты труда	55842,56
Единый социальный налог (26,2%)	14630,75
Увеличение стоимости основных средств (2 компьютера по цене 7500 грн.)	15000,00
Амортизация программно-аппаратного комплекса	3000,00
Командировочные расходы (Киев – 5 дней)	5750,00
Коммунальные услуги, услуги связи (телефон, Интернет) (750 грн.×12 мес.)	9000,00
Прочие расходы (250 грн. × 12 мес.)	3000,00
Итого прямые расходы	96973,31
Фонд развития производства (10% от прямых затрат)	9697,33
Накладные расходы (12% от прямых затрат)	11636,80
Всего расходов	118307,44
Налог на добавленную стоимость (18% от общей стоимости проекта)	21295,34
ИТОГО ДОГОВОРНАЯ ЦЕНА	139602,78

4 Расчет точки безубыточности

Расчет точки безубыточности выполняется в тех случаях, когда разработка является инициативной и выполняется за счет средств разрабатывающей организации.

Исследовав рынок программного обеспечения и цены конкурентов, экспертами отдела маркетинга установлено, что рекомендуемая стоимость *C* одной копии системы будет составлять порядка **6500 гривень.** Основная зарплата специалистов отдела маркетинга (зав. отделом, программист, маркетолог, экономист) по нормам, установленным в разрабатываемой организации, составляет 34.6% от стоимости тиражируемого продукта (**2250 грн**.. в месяц). Фиксированными издержками являются (таблица 28):

Таблица 28.

Постоянные (фиксированные) расходы F_c

Наименование расходов	Сумма (грн.)
Накладные расходы по проекту (10%)	950
Плановое ежемесячное гашение кредита (140000 грн./12 мес.)	11670
Выплата среднего банковского процента - 25% годовых	2920
Амортизация программно-аппаратного комплекса отдела маркетинга (2 компьютера по 6 тыс. грн., лаз. принтер – 2,5 тыс. грн.)	260
Прочие расходы	200
ИТОГО	16000

Переменные издержки отдела маркетинга, занимающегося непосредственно тиражированием программного продукта, рассчитываются на единицу продукции (таблица 29).

Объем выпуска, при котором достигается точка безубыточности (нулевой уровень прибыли) определяется по формуле 16:

Таблица 29.

Переменные издержки (отдел маркетинга) V_c

Наименование расходов	Сумма (руб.)
Основная зарплата специалистов (34,6 % от стоимости тиражируемого продукта)	2250
Единый социальный налог (26,2%)	589,5
Комплектующие и расходные материалы (картриджи, тонер, бумага, диски CD-DVD и т.д.)	99,5
Накладные расходы отдела маркетинга (транспорт, услуги связи, Интернет, телефоны и т.д.)	60
ИТОГО	3000

$$\nu_0 = \frac{F_c}{C - V_c} = \frac{16000}{6500 - 3000} = 4{,}57$$

Период, за который был реализован данный объем программной продукции, определяется сроком, за который мы определили величину постоянных издержек – в данном случае 1 месяц.

Таким образом, в течение месяца фирме необходимо подготовить и продать 5 копий программного обеспечения по цене 6500 грн., чтобы покрыть постоянные и переменные расходы по ее деятельности.

5. Расчет технико-экономической эффективности разработки ПС

Расчет технико-экономической эффективности разработки ПС выполняется в тех случаях, когда на предприятии уже эксплуатируется система, выполняющая аналогичные разрабатываемой ПС функции или в случаях, когда на рынке ПС имеются подобные системы.

5.1. Расчет коэффициента технического уровня

В качестве программы-аналога при разработке проекта принята эксплуатирующаяся в настоящее время на предприятии программа «Склад». Показатели качества разрабатываемой ПС и аналога, оцененные экспертом, приведены в таблице 30.

Таблица 30.

Показатели качества

Показатели качества	Весовой коэффициент B_i	Проект		Аналог	
		X_i^1	$B_i \times X_i^1$	X_i^2	$B_i \times X_i^2$
Удобство работы (пользовательский интерфейс)	0,1	4	0,4	2	0,2
Новизна (соответствие современным требованиям)	0,06	4	0,24	3	0,18
Соответствие профилю деятельности заказчика	0,15	4	0,6	2	0,3
Операционная система (многозадачность, графика)	0,05	4	0,2	4	0,2
Надежность (защита данных)	0,13	3	0,39	3	0,39
Скорость доступа к данным	0,09	4	0,36	4	0,36
Гибкость	0,05	3	0,15	3	0,15
Функции обработки информации	0,13	5	0,65	1	0,13
Соотношение стоимость/возможности	0,09	4	0,36	2	0,18
Время обучения персонала	0,15	4	0,6	2	0,3
Обобщенный показатель качества J		J_1=3,95		J_2=2,39	

Коэффициент технического уровня в соответствии с (20) составляет

$$A = \frac{J_1}{J_2} = \frac{3{,}95}{2{,}39} = 1{,}65$$

Так как коэффициент больше 1, то разработка проекта с технической точки зрения оправдана.

5.2 Смета эксплуатационных затрат

Эксплуатацию разработанной системы осуществляют специалисты - сотрудник отдела материально-технического снабжения и программист. Заработная плата специалистов определяется пропорционально степени участия каждого из них в эксплуатации ПС. Результаты расчета годовой заработной платы для разрабатываемой ПС и аналога приведены в таблице 31. Базовый оклад программиста принят 3750 грн./мес., сотрудника МТС 2500 грн./мес.

Таблица 31

Заработная плата специалистов

Должность	Сред .днев. ставка (грн./день)	Проект		Аналог	
		Затраты времени (чел. дней)	ФОТ грн.	Затраты времени (чел. дней)	ФОТ грн.
Сотрудник МТС	104,2	40	4168	40	4168
Программист	156,25	20	3125	60	9375
Итого		7293		13543	

Общие годовые эксплуатационные затраты по проекту и аналогу приведены в таблице 32.

Таблица 32

Годовые эксплуатационные затраты

Статьи затрат	Затраты	
	Проект	Аналог
Фонд оплаты труда	7291,67	13541,67
Единый социальный налог (26,2% от ФОТ)	1910,42	3547,92
Амортизационные отчисления	3000	3000
Затраты на электроэнергию	100	100
Затраты на текущий ремонт	150	150
Затраты на материалы	100	100
Накладные расходы (18% от прямых затрат)	2239,13	3658,88
ИТОГО	14791,22	24098,47

В соответствии с (22) рассчитываем приведенные затраты на единицу работ:

$$W_1 = C_1 + E_n \times K_1 = 14791{,}22 + 0{,}33 \times 140000 = 60991{,}22 грн..$$

$$W_2 = C_2 + E_n \times K_2 = 24098{,}47 + 0{,}33 \times 197500 = 89273{,}47 грн.$$

где в качестве K_2 взята стоимость разработки аналога 197500 грн.

Отсюда, в соответствии с (21), экономический эффект от внедрения разработанной программной системы составит

$$E = W_2 \times A - W_1 = 89273{,}47 \times 1{,}65 - 60991{,}22 = 86310{,}01 грн.$$

Срок окупаемости

$$T_0 = \frac{K_1}{E} = \frac{140000}{86310{,}01} = 1{,}63 года$$

Фактический коэффициент экономической эффективности равен

$$E_f = \frac{1}{T_0} = \frac{1}{1{,}63} = 0{,}62$$

Фактический коэффициент экономической эффективности разработки получился больше, чем нормативный, поэтому разработка и внедрение разрабатываемого продукта является эффективной.

Выводы

В таблице32 приведены сводные экономические характеристики проекта

Таблица 32

Характеристики проекта	**Значения**
Затраты на разработку проекта, грн.	140000
Общие эксплуатационные затраты, грн.	14791,22
Экономический эффект, грн.	86310,01
Коэффициент экономической эффективности	0,62
Срок окупаемости, лет	1,63
Количество реализаций за месяц, при продажной цене 6500 грн.	5

ЛИТЕРАТУРА

1. ГОСТ ИСО/МЭК 12119-2000 «Информационная технология. Пакеты программ. Требование к качеству и тестирование».
2. ГОСТ 28195-89 »Оценка качества программных средств. Общие положения».
3. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. — М.: Издательский дом «Вильямс», 2004г. – 1136 с.
4. Международный стандарт ISO/IEC 9126-1:2001. Программирование. Качество продукта. Часть 1. Модель качества.
5. Липаев В. В. Технико-экономическое обоснование проектов сложных программных систем. — М.: СИНТЕГ, 2004, - 284 с.
6. ГОСТ 34.003-90 Информационная технология. Автоматизированные системы. Термины и определения.
7. Зельковиц М., Шоу А., Гэннон Дж. Принципы разработки программного обеспечения. – М.: Мир, 1982 – 368 с.