

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**\_Б1.В.ДВ.07.01 Экономика и менеджмент в информационной безопасности  
критически важных объектов**

**Специальность** 10.05.03 Информационная безопасность автоматизированных систем

**Специализация** Информационная безопасность автоматизированных систем критически  
важных объектов

**Форма обучения** очная

## СОДЕРЖАНИЕ

1. Конспект лекций .....	3
1.1. Лекция № 1 Основные понятия экономики и менеджмента информационной безопасности критически важных объектов.	3
1.2. Лекция № 2 Прямой метод определения технико-экономических показателей проекта.....	6
1.3. Лекция № 3 Определение технико-экономических показателей подсистемы защиты критически важных объектов с использованием метода функциональных точек.	8
1.4. Лекция № 4 ) Определение технико-экономических показателей проекта на основе размерности базы данных .....	10
1.5. Лекция № 5 Определение договорной цены на разработку подсистемы защиты критически важных объектов. Определение фонда оплаты труда на разработку подсистемы защиты критически важных объектов.	11
1.6. Лекция № 6 Определение договорной цены на разработку подсистемы защиты критически важных объектов. Определение фонда оплаты труда на проведение опытной эксплуатации подсистемы защиты критически важных объектов.	12
1.7. Лекция № 7 Определение договорной цены на разработку подсистемы защиты критически важных объектов. Структура договорной цены на подсистемы защиты критически важных объектов.	13
1.8. Лекция № 8 Анализ рыночной стоимости подсистемы защиты критически важных объектов.	14
2. Методические указания по проведению практических занятий .....	18
2.1 <i>Практическое занятие 1</i> Основные понятия экономики информационной безопасности критически важных объектов.	
2.2 <i>Практическое занятие 2</i> Основные понятия менеджмента информационной безопасности критически важных объектов..	
2.3 <i>Практическое занятие 3</i> Основы описания программного обеспечения как объекта продажи и поставки	
2.4. <i>Практическое занятие 4.</i> Оценка качества программных средств по ГОСТ 28195-89.....	19
2.5. <i>Практическое занятие 5.</i> Декомпозиция системы защиты критически важных объектов для оценки каждого из компонент системы.	20
2.6 <i>Практическое занятие 6.</i> Определение трудозатрат, длительности реализации проекта и средней численности разработчиков для прямого метода.....	21
2.5. <i>Практическое занятие 7.</i> Представление подсистемы защиты критически важных объектов в виде многоуровневой графической модели. Расчет количества функциональных точек.	22
2.8 <i>Практическое занятие 8.</i> Определение трудозатрат, длительности и средней численности специалистов на основе базовой конструктивной модели трудозатрат при использовании метода функциональных точек.....	24
2.9. <i>Практическое занятие 9.</i> Определение размерности подсистемы защиты критически важных объектов по количеству объектов, атрибутов и их взаимосвязям в базе данных подсистемы защиты критически важных объектов.	25
2.10. <i>Практическое занятие 10.</i> Расчет трудозатрат на разработку базы данных большой размерности и длительности ее разработки.....	26
2.11 <i>Практическое занятие 11.</i> Расчет трудозатрат и длительности по основным этапам жизненного цикла создания подсистемы защиты критически важных объектов.	27
2.12 <i>Практическое занятие 12.</i> Определение численности каждого типа	27

специалистов на каждом этапе жизненного цикла создания подсистемы защиты критически важных объектов. Расчет общего фонда заработанной платы на реализацию проекта подсистемы защиты критически важных объектов.	
2.13 <i>Практическое занятие 13.</i> Перечень работ на этапе опытной эксплуатации разрабатываемой подсистемы защиты критически важных объектов согласно ГОСТ 34.603-92. Нормативы трудоемкости опытной эксплуатации подсистемы защиты критически важных объектов.	28
2.14 <i>Практическое занятие 14.</i> Расчет фонда заработанной платы на проведение опытной эксплуатации разрабатываемой подсистемы защиты критически важных объектов.	29
2.15 <i>Практическое занятие 15.</i> Составление сметы затрат на разработку и внедрение подсистемы защиты критически важных объектов.	29

# **1. КОНСПЕКТ ЛЕКЦИЙ**

## **1.1. Лекция № 1 (2 часа)**

**Тема:** «Основные понятия экономики и менеджмента информационной безопасности критически важных объектов.»

### **1.1.1. Вопросы лекции:**

- 1) Понятие технико-экономического обоснования стоимости АСОИ.
- 2) Основные показатели оценки экономической эффективности разрабатываемой АСОИ.
- 3) Типы и основные требования к АСОИ.
- 4) Алгоритм подготовки ТЭО договорной цены АСОИ.

### **1.1.2. Краткое содержание вопросов:**

#### **1) Понятие технико-экономического обоснования стоимости АСОИ**

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

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.

#### **2) Основные показатели оценки экономической эффективности разрабатываемой АСОИ.**

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 будем понимать 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 предметной области и разработка требований к программной системе;

- проектирование;
- программирование;
- 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) Типы и основные требования к АСОИ.

По уровню сложности все множество программных систем (ПС) следует 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 информационно-справочные системы (ИСС), обеспечивающие информационную поддержку основных бизнес-процессов организации с большим количеством типов исходной информации.

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

При формировании требований к программной системе необходимо использование стандарта ISO/IEC 9126-1:2001 [3].

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.

4) Алгоритм подготовки ТЭО договорной цены АСОИ.

1. Составляется описание программного продукта (пример – в приложении 1).

2. Определяется тип системы и сложность системы (раздел 2.1.2.2.).

3. Фиксируется язык программирования.

4. Выбираются (задаются преподавателем-консультантом) два из трех методов определения технико-экономических параметров программной системы, которые необходимо применить при подготовке экономической части ДП:

- прямой метод определения технико-экономических показателей проекта (метод экспертизы оценок);

- метод функциональных точек;

- метод определения ТЭП проекта на основе размерности базы данных ПС.

5. Задается плановый срок на разработку системы, установленный заказчиком.

6. При работе с прямым методом определения ТЭП проекта:

1) производится анализ объекта автоматизации (декомпозиция бизнес-процессов) и определяется состав программных комплексов и компонент ПС;

2) определяется (задается) количество экспертов, определяющих размерность программной системы;

3) эксперты заполняют бланки экспертного оценивания размерности программной системы;

4) с помощью формул 2.1 и 2.2 вычисляется размерность программной системы;

5) проводится оценка трудозатрат и средней численности разработчиков при реализации проекта (табл. 2.2, формула 2.3.).

7. При работе с методом определения ТЭП *на основе размерности базы данных программной системы*:

1) производится анализ объекта автоматизации (ER-моделирование) и строится физическая (логическая, концептуальная) модель базы данных ПС;

2) определяются параметры полученной БД, необходимые для расчета размерности ПС (количество объектов, взаимосвязей и атрибутов);

3) определяется размерность программного обеспечения (в данном случае – базы данных) – формула 2.15;

4) определяются трудозатраты и средняя численность разработчиков при реализации проекта, при этом трудозатраты определяются по формуле 2.16 на основе статистических нормативов трудоемкости разработки программной системы, приведенных в табл. 2.15 (раздел 2.3.3).

8. При определении ТЭП с помощью *метода функциональных точек*:

1) выделяется множество бизнес-процессов;

2) производится подсчет количества функциональных точек бизнес-процесса в разрезе каждой категории функций (простые/средние/сложные) для заданной последовательности этапов (ввод, вывод, опросы, структуры данных, интерфейсы) – таблицы 2.3-2.9, раздел 2.3.2 – формула 2.5;

3) производится учет факторов и требований среды разработки программной системы (табл. 2.10, 2.11) – формула 2.6;

4) по формуле 2.7 рассчитывается уточненное количество функциональных точек с учетом факторов внешней среды;

5) по формуле 2.8 определяется размерность программного обеспечения для конкретного языка программирования с учетом нормативов, представленных в таблице 2.12;

6) производится определение трудозатрат при заданной длительности и средней численности специалистов на основе базовой конструктивной модели трудозатрат – СОСМО: трудозатраты определяются по формуле 2.9, таблица 2.13;

7) по формуле 2.10 определяется средняя численность сотрудников, занятых в проекте.

## 1.2. Лекция № 2 (2 часа)

**Тема:** «Прямой метод определения технико-экономических показателей проекта».

### 1.2.1. Вопросы лекции:

- 1) Метод экспертных оценок.
- 2) Декомпозиция АСОИ для оценки каждого из компонент системы.
- 3) Определение трудозатрат, длительности реализации проекта и средней численности разработчиков.
- 4) Алгоритм определения ТЭП АСОИ прямым методом.

### 1.2.2. Краткое содержание вопросов:

- 1) Метод экспертных оценок.

Прямой метод определения технико-экономических параметров программной системы основан на использовании **метода экспертных оценок**.

Будущую программную систему следует декомпозировать до уровня элементарных компонент, а для оценки размеров каждого из компонент использовать либо внешних экспертов, имеющих опыт разработки подобных систем и готовые прототипы, либо, в качестве экспертов могут выступать специалисты разработчика и заказчика.

- 2) Декомпозиция АСОИ для оценки каждого из компонент системы.

При декомпозиции целесообразно использовать следующие термины и определения, изложенные в [5] (рис. 2.1):

- **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 бизнес-процессов организации;
- **программный комплекс** – совокупность программных компонент, реализующих конкретный 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.

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.

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

Средняя оценка по бета-распределению определяется путем умножения реалистической оценки на 4, добавлением оптимистической и пессимистической оценок и делением полученного результата на 6.

3) Определение трудозатрат, длительности реализации проекта и средней численности разработчиков.

Оценка трудозатрат, длительности и средней численности разработчиков при реализации проекта основывается на согласовании между разработчиком и заказчиком производительности труда программиста - Р.

В [4] приводятся следующие среднестатистические оценки производительности труда программиста:

- при разработке программных систем первого класса сложности (КПС) преимущественно на языке ассемблер – 60-80 строк/чел.-месяц;

- при разработке программных систем второго класса сложности (ИСС) на языках высокого уровня – 250-260 строк/чел.-месяц.

В таблице 2.2. представлены статистические показатели производительности, рекомендуемые в базовой модели Constructive Cost Model.

COnstructive COst MOdel (COCOMO – модель издержек разработки) – это алгоритмическая модель оценки стоимости разработки программного обеспечения, разработанная Барри Боэмом (Barry Boehm). Модель использует простую формулу регрессии с параметрами, определенными из данных, собранных по ряду проектов.

COCOMO была впервые опубликована в 1981 году в книге Боэма «Экономика разработки программного обеспечения» в качестве модели для оценки трудоемкости, себестоимости и плана-графика для проектов по разработке ПО.

Она использовала исследование 63 проектов в аэрокосмической компании TRW, в которой Боэм был директором отдела исследований программного обеспечения и технологий.

В исследовании проекты классифицировались по размеру в зависимости от количества строк кода (от 2 до 100 тысяч), а также по языку программирования (от ассемблеров до высоко-уровневого языка PL/1).

В 1997 году была разработана модель COCOMO II, которая стала наследником первоначальной модели и более подходящей для оценивания современных проектов разработки ПО. Она предоставляет более полную поддержку современных процессов разработки ПО и построена на обновленной базе проектов.

В целом, COCOMO состоит из иерархии трех последовательно детализируемых и уточняемых форм.

Первый уровень – **Базовый**, хорош для быстрых ранних оценок стоимости разработки ПО, однако не принимает во внимание различия в аппаратных ограничениях, качестве и опыте персонала, а также использованию современных техник и средств разработки и других факторов, которые невозможно учесть на ранних стадиях разработки.

**Средний** уровень COCOMO учитывает эти факторы, тогда как **Детальный** уровень дополнительно учитывает влияние от дальних фаз проекта на его общую стоимость.

Средний уровень (COCOMO Model 2: Intermediate) – рассчитывает трудоемкость разработки как функцию от размера программы и множества «факторов стоимости», включающих субъективные оценки характеристик продукта, проекта, персонала и аппаратного обеспечения. Детальный уровень (COCOMO Model 3: Advanced/Detailed) – включает в себя все характеристики среднего уровня с оценкой влияния данных характеристик на каждый этап процесса разработки ПО.

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

программного комплекса. С учетом вышеизложенного, трудозатраты на разработку системы могут быть определены следующим образом:

$$T = R/P \quad (2.3)$$

4) Алгоритм определения ТЭП АСОИ прямым методом.

При работе с **прямым методом** определения ТЭП проекта:

1) производится анализ объекта автоматизации (декомпозиция бизнес-процессов) и определяется состав программных комплексов и компонент ПС;

2) определяется (задается) количество экспертов, определяющих размерность программной системы;

3) эксперты заполняют бланки экспертного оценивания размерности программной системы

4) с помощью формул 2.1. и 2.2. вычисляется размерность программной системы;

5) проводится оценка трудозатрат и средней численности разработчиков при реализации проекта

### 1.3. Лекция № 3 (2 часа)

**Тема:** «Определение технико-экономических показателей подсистемы защиты критически важных объектов с использованием метода функциональных точек»

#### 1.3.1. Вопросы лекции:

1) Представление АСОИ в виде многоуровневой графической модели.

2) Расчет количества функциональных точек.

3) Определение трудозатрат, длительности и средней численности специалистов на основе базовой конструктивной модели трудозатрат.

4) Алгоритм определения ТЭП АСОИ методом функциональных точек.

#### 1.3.2. Краткое содержание вопросов:

1) Представление АСОИ в виде многоуровневой графической модели.

**Метод функциональных точек** (Function point – FP) основывается на том, что размеры программной системы оцениваются в терминах количества и сложности бизнес-процессов (функций), реализуемых в данном программном коде [2].

**Функциональная точка** – это комбинация свойств программного обеспечения:

- интенсивности использования ввода и вывода внешних данных;
- взаимодействия системы с пользователем;
- внешних интерфейсов;
- файлов, используемых системой.

Будущая система с использованием методологии структурного анализа и проектирования описывается в виде многоуровневой графической модели, представленной в виде совокупности взаимосвязанных функциональных диаграмм (пользовательских бизнес-процессов).

Каждый из бизнес-процессов включает в себя входные и выходные данные, преобразования и внешние интерфейсы.

Процедура оценивания размеров программной системы соотносится с одним из пользовательских бизнес-процессов и состоит из следующей последовательности этапов (ввод, вывод, опросы, структуры данных, интерфейсы):

- выделение множества бизнес-процессов;
- подсчет количества функциональных точек бизнес-процесса в разрезе каждой категории;
- определение весовых коэффициентов сложности каждой функции;
- учет факторов и требований среды разработки программной системы;
- вычисление интегральных показателей сложности;
- вычисление итогового количества функциональных точек;
- определение размеров программного комплекса в показателях LOC;

- 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;
- при декомпозиции сложной функции учитываются все логические преобразования с данными.

## 2) Расчет количества функциональных точек.

**Определение количества выводов.** Под выводами будем понимать следующие единицы информации, получаемые на выходе рассматриваемого бизнес-процесса:

- файлы, продуцируемые в данном бизнес-процессе 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.

Каждый из выводов, в зависимости от количества файлов, используемых при формировании выходов, рекомендуется отнести к одной из категорий сложности: простой, средний, сложный.

**Определение количества вводов.** Под вводами будем понимать следующие единицы информации, поступающие на вход рассматриваемого бизнес-процесса:

- 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

**Определение количества опросов ввода, вывода.** Под опросами будем понимать следующие действия, исполняемые программной системой в рассматриваемом бизнес-процессе:

- обращение к внешним процедурам, оформленным в виде специфических 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.

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.

**Определение количества файлов.** Под файлами будем понимать следующие единицы информации, использующиеся программной системой в рассматриваемом бизнес-процессе:

- внутренние логические файлы программной системы;
- структуры данных, представляющие собой первичную логическую группу пользовательских данных, которые постоянно находятся внутри границ программной системы;
- внешние файлы, доступные пользователям с помощью ввода, вывода, опросов, либо интерфейсов.

## 3) Определение трудозатрат, длительности и средней численности специалистов на основе базовой конструктивной модели трудозатрат.

В основу *оценки трудозатрат* положена степенная функция следующего вида:

$$T = A * R^E (KLOC) / 12, \quad (2.9) \text{ где } T - \text{трудозатраты, выраженные в человеко-месяцах;}$$

$R (KLOC)$  – размерность программной системы, выраженная в тысячах строк кода.

Первый сомножитель  $A$  является доминирующим, он прямо пропорционален размерности программного обеспечения  $R$  и отражает линейную зависимость роста трудозатрат от размерности.

Второй сомножитель  $R^E$  отражает тот факт, что при увеличении размерности программной системы возрастает относительная трудоемкость разработки каждой строки программного кода за счет увеличения количества взаимосвязей между компонентами.

Значения параметров  $A$  и  $E$ , полученные путем статистической обработки данных по результатам реализации множества проектов, представлены в табл. 2.13 [4].

Оценки по СОСОМО получены в результате обработки статистических данных по 160 реальным зарубежным проектам, а оценки по ПРОМЕТЕЮ – результат обобщения статистики – по 250 отечественным.

**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:

$$Z = T / D \quad (2.10)$$

Расчет длительности и, соответственно, численности специалистов для разработки программного обеспечения может быть произведен также из среднестатистической производительности труда программиста.

4) Алгоритм определения ТЭП АСОИ методом функциональных точек.

При определении ТЭП с помощью метода функциональных точек:

1) выделяется множество бизнес-процессов;

2) производится подсчет количества функциональных точек бизнес-процесса в разрезе каждой категории функций (простые/средние/сложные) для заданной последовательности этапов (ввод, вывод, опросы, структуры данных, интерфейсы) – таблицы 2.3-2.9, раздел 2.3.2 – формула 2.5;

3) производится учет факторов и требований среды разработки программной системы (табл. 2.10, 2.11) – формула 2.6;

4) по формуле 2.7. рассчитывается уточненное количество функциональных точек с учетом факторов внешней среды;

5) по формуле 2.8. определяется размерность программного обеспечения для конкретного языка программирования с учетом нормативов, представленных в таблице 2.12;

6) производится определение трудозатрат при заданной длительности и средней численности специалистов на основе

#### 1.4. Лекция № 4 (2 часа)

**Тема:** «Определение технико-экономических показателей проекта на основе размерности базы данных».

##### 1.4.1. Вопросы лекции:

1) Определение размерности АСОИ по количеству объектов, атрибутов и их взаимосвязям в базе данных АСОИ.

2) Расчет трудозатрат на разработку базы данных большой размерности и длительности ее разработки.

3) Алгоритм определения ТЭП АСОИ на основе размерности базы данных.

#### **1.4.2. Краткое содержание вопросов:**

1) Определение размерности АСОИ по количеству объектов, атрибутов и их взаимосвязям в базе данных АСОИ.

Размерность программной системы определяется количеством объектов, атрибутов и их взаимосвязями на объектных диаграммах бизнес-процессов [6].

**Атрибут** – простейший элемент базы данных информационной модели, содержащей одну из характеристик предметной области и вводимой либо непосредственно пользователем, либо анонсирующей 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:

$R = 2N * 5K^I * 10M$ , (2.14) где  $N$  – количество объектов (таблиц) предметной области, количество связей между таблицами неограниченно и определяется структурой базы данных;

$K1$  – суммарное количество взаимосвязей между объектами;

$M$  – суммарное количество атрибутов предметной области, **приходящихся на один объект**.

2) Расчет трудозатрат на разработку базы данных большой размерности и длительности ее разработки.

В модифицированной модели СОСМО II (средний уровень) при определении трудоемкости **учитываются дополнительно пять групп факторов**, влияющих на технико-экономические показатели проекта:

- масштабность проекта;
- требования к показателям качества программного обеспечения;
- квалификация коллектива разработчиков;
- характеристики технологической среды разработки;
- характеристики программно-аппаратной среды разработки.

В табл. 2.14 представлен перечень конкретных показателей по каждой из вышеперечисленных групп и их максимальные значения. Конкретные значения показателей определяются совместно представителями разработчика и заказчика. При значении показателя равным **единице** считается, что соответствующий фактор не влияет на трудоемкость разработки программной системы.

3) Алгоритм определения ТЭП АСОИ на основе размерности базы данных.

При работе с методом определения ТЭП **на основе размерности базы данных программной системы**:

1) производится анализ объекта автоматизации (ER-моделирование) и строится физическая (логическая, концептуальная) модель базы данных ПС;

2) определяются параметры полученной БД, необходимы для расчета размерности ПС (количество объектов, взаимосвязей и атрибутов);

3) определяется размерность программного обеспечения (в данном случае – базы данных) – формула 2.15;

4) определяются трудозатраты и средняя численность разработчиков при реализации проекта, при этом трудозатраты определяются по формуле 2.16 на основе статистических нормативов трудоемкости разработки программной системы, приведенных в табл. 2.15.

## **1.5. Лекция № 5 (2 часа)**

**Тема:** «Определение договорной цены на разработку подсистемы защиты критически важных объектов. Определение фонда оплаты труда на разработку подсистемы защиты критически важных объектов».

### **1.5.1. Вопросы лекции:**

1) Расчет трудозатрат и длительности по основным этапам жизненного цикла создания АСОИ.

2) Определение численности каждого типа специалистов на каждом этапе жизненного цикла создания АСОИ.

3) Расчет общего фонда заработанной платы на реализацию проекта АСОИ.

### **1.5.2. Краткое содержание вопросов:**

1) Расчет трудозатрат и длительности по основным этапам жизненного цикла создания АСОИ.

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.

Используя эти распределения, по выражению 2.17 можно рассчитать среднюю численность сотрудников, занятых на каждом из этапов создания программной системы.

$$Z T \Delta i i = \alpha / \beta, i=1,4 \quad (2.17)$$

2) Определение численности каждого типа специалистов на каждом этапе жизненного цикла создания АСОИ.

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

$Z = P \cdot Z, i=1,4 j=1,3, \quad (2.18)$  где  $P$  – относительная доля (%) специалистов  $j$ -го типа, привлекаемых для реализации проекта на  $i$ -ом этапе.

3) Расчет общего фонда заработанной платы на реализацию проекта АСОИ.

Фонд заработной платы для реализации  $i$ -го этапа проекта определяется по формуле:  $S_i = \sum Z_{ij} * \Delta_j * S_j, i=1,4 \quad (2.19)$  где  $\Delta_i$  – длительность  $i$ -го этапа проекта;  $S_j$  – месячный фонд заработной платы специалиста  $j$ -го типа.

В основу определения  $S_j$  может быть положена месячная базовая ставка программиста, размер которой может быть принят как одна из альтернатив: базовая ставка программиста заказчика; базовая ставка программиста разработчика; средне - рыночная базовая ставка программиста в данном регионе.

В качестве ориентира для согласования базовой ставки может быть принята стоимость разработки одной строки исходного кода программы 20 -50 \$ US [4] и принятой производительности труда (раздел 2.3.1.).

При этом месячная ставка специалиста-программиста соотносится к месячной ставке системного аналитика как 1:1,3, а к ставке технического специалиста – как 1:0,7.

Общий фонд заработной платы на реализацию проекта определяется по формуле:

$$S = \sum_i S_i \quad (2.20)$$

$i = 1$

## **1.6. Лекция № 6 (2 часа)**

**Тема:** «Определение договорной цены на разработку подсистемы защиты критически важных объектов. Определение фонда оплаты труда на проведение опытной эксплуатации подсистемы защиты критически важных объектов».

### **1.6.1. Вопросы лекции:**

1) Перечень работ на этапе опытной эксплуатации разрабатываемой АСОИ согласно ГОСТ 34.603-92.

2) Нормативы трудоемкости опытной эксплуатации АСОИ.

3) Расчет фонда заработанной платы на проведение опытной эксплуатации разрабатываемой АСОИ.

#### **1.6.2. Краткое содержание вопросов:**

1) Перечень работ на этапе опытной эксплуатации разрабатываемой АСОИ согласно ГОСТ 34.603-92.

На этапе опытной эксплуатации программной системы в соответствии с ГОСТ 34.603-92 производится заполнение нормативной базы, справочников, классификаторов, эксплуатация программного обеспечения в регламентном режиме, доработка программного обеспечения и рабочей документации в случае несоответствия текущей версии системы требованиям технического задания. Срок опытной эксплуатации оговаривается в техническом задании.

Численность сотрудников, привлекаемых к опытной эксплуатации определяется по формуле:

$Z_{оп} = t_{оп} \cdot N$ , (2.21) где  $t_{оп}$  — срок опытной эксплуатации;  $N$  — норматив трудоемкости при проведении опытной эксплуатации.

2) Нормативы трудоемкости опытной эксплуатации АСОИ.

Нормативы трудоемкости опытной эксплуатации программной системы определяются исходя из среднестатистических нормативов.

Опытная эксплуатация проводится группой внедрения разработчика с привлечением, в случае необходимости, программистов, при этом относительный норматив заработной платы специалиста составляет 0,85.

3) Фонд заработной платы на проведение опытной эксплуатации определяется по формуле:  $S_{оп} = Z_{оп} \cdot t_{оп} \cdot S_p \cdot 0,85$ , (2.22), где  $S_p$  — месячная базовая ставка программиста.

### **1.7. Лекция № 7 (2 часа)**

**Тема:** «Определение договорной цены на разработку подсистемы защиты критически важных объектов. Структура договорной цены на подсистемы защиты критически важных объектов».

#### **1.7.1. Вопросы лекции:**

1) Прямые и накладные расходы на разработку АСОИ.

2) Фонд развития производства. Налог на добавленную стоимость.

3) Особенности составления сметы затрат на разработку и внедрение АСОИ.

#### **1.7.2. Краткое содержание вопросов:**

1) Прямые и накладные расходы на разработку АСОИ.

Структура договорной цены представлена в табл. 2.19 и состоит из следующих разделов:

##### *1. Прямые расходы:*

- фонд оплаты труда;

- страховые взносы в Пенсионный фонд (ПФР), Фонд социального страхования (ФСС) и Федеральный фонд обязательного медицинского страхования (ФОМС);

- увеличение стоимости основных средств – по данной статье приобретаются все основные средства организации (в т.ч. средства вычислительной техники);

- амортизация средств ВТ стоимостью от 40000 рублей и выше, используемых разработчиком при реализации проекта, исходя из установленного в организации срока полезного использования техники (25-36 месяцев);

- командировочные расходы – при необходимости (определяются на договорной основе), а для бюджетных учреждений
  - на основании Приказов Минфина РФ;
  - оплата коммунальных услуг, размер которых определяется исходя из численности специалистов и нормативов площади на 1 рабочее место;
  - прочие расходы (расходные материалы, услуги связи и т.д.) определяются на договорной основе – по согласованию с заказчиком, при этом услуги связи (телефоны, Интернет) определяются действующими расценками со стороны региональных отделений ОАО «Ростелеком» и провайдеров сети Интернет;

2. *Накладные расходы* организации-разработчика (расходы на АУП, охрану, обслуживающий персонал и т.д.), размер которых подтверждается соответствующими документами;

2) Фонд развития производства. Налог на добавленную стоимость.

1. *Налог на добавленную стоимость* определяется Налоговым кодексом РФ и формой организации предприятия- разработчика. Следует при этом учесть, что согласно налоговому кодексу РФ не подлежат налогообложению (освобождаются от НДС) следующие виды работ:

- выполнение научно-исследовательских и опытно-конструкторских работ за счет средств бюджетов, а также средств Российского фонда фундаментальных исследований,

Российского фонда технологического развития и образуемых для этих целей в соответствии с законодательством РФ внебюджетных фондов министерств, ведомств, ассоциаций;

- выполнение НИР и ОКР учреждениями образования и науки на основе хозяйственных договоров.

3) Особенности составления сметы затрат на разработку и внедрение АСОИ.

Структура договорной цены

1. Наименование статей расходов:

1) Оплата труда непосредственных исполнителей.

2) Страховые взносы в ПФР, ФСС, Федеральный ФОМС.

3) Увеличение стоимости основных средств.

4) Амортизация средств вычислительной техники стоимостью от 40000 рублей и выше.

5) Прочие выплаты (командировки в части суточных).

6) Транспортные услуги (командировки в части оплаты транспортных расходов).

7) Прочие услуги (командировки в части проживания, оплата услуг сторонних организаций по НИР, ОКР и технологическим работам, договора подряда с ЕЧН).

8) Прочие расходы (расходные материалы, услуги связи и т.д.).

9) Накладные расходы организации разработчика (расходы на АУП, охрану, обслуживающий персонал и т.д.).

10) Фонд развития производства.

11) Налог на добавленную стоимость.

## **1.8. Лекция № 8 (2 часа)**

**Тема:** «Анализ рыночной стоимости подсистемы защиты критически важных объектов.».

### **1.8.1. Вопросы лекции:**

- 1) Концепция безубыточности.
- 2) Виды и составляющие издержек.
- 3) Определение точки безубыточности.

### **1.8.2. Краткое содержание вопросов:**

- 1) Концепция безубыточности.

История успеха любой компании измеряется величиной полученной прибыли и изменением ее динамики. Для успешного ведения бизнеса необходимо не только просчитывать, сколько компания заработает при достижении запланированного объема продаж, но и четко представлять, какой минимальный объем продаж необходим для обеспечения безубыточной работы.

Задача стоит в определении того объема продаж, ниже которого предприятие будет терять деньги, выше которого – зарабатывать. Этот минимально допустимый объем продаж, который покрывает все затраты на изготовление продукции, не принося при этом ни прибыли, ни убытков, получил название **точка безубыточности** (она же – точка равновесия, она же – break-even point).

С точки зрения экономической теории безубыточность есть нормальное состояние фирмы на современном конкурентном рынке, находящемся в состоянии долгосрочного равновесия. При этом мы принимаем в рассмотрение экономическую прибыль, то есть то определение прибыли, при котором в расходы фирмы включается среднерыночная ставка дохода на инвестированный капитал, а также нормальный доход предприятия.

При таких допущениях определение безубыточности звучит следующим образом: «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».

В этом случае, если фирма имеет бухгалтерскую прибыль, то есть сальдо доходов от продаж и денежных затрат на производство проданной продукции положительно и при этом прибыль фирмы меньше, чем среднерыночный процент на собственный капитал фирмы, то она может не достичь точки безубыточности в смысле экономической прибыли. Следовательно, существуют более выгодные способы использования капитала, которые позволяют получить более высокую прибыль.

Таким образом, понятие точки безубыточности является одновременно и неким критерием эффективности деятельности фирмы. Фирма, не достигающая точки безубыточности действует неэффективно с точки зрения сложившейся рыночной конъюнктуры. Однако этот факт сам по себе не служит однозначной причиной для прекращения существования фирмы. Для того, чтобы ответить на этот вопрос необходимо детально исследовать структуру издержек фирмы.

Из курса микроэкономической теории известно, что начиная с некоторого объема выпуска кривая переменных издержек будет являться возрастающей, а кривая предельного дохода –убывающей. Для целей максимизации прибыли основным является соотношение между предельным доходом и предельными издержками при увеличении выпуска на одну единицу. Очевидно, что в случае, когда предельные издержки меньше предельного дохода, увеличение выпуска повлечет за собой увеличение дохода фирмы. В случае же когда предельные издержки больше предельного дохода, к увеличению предельного дохода фирмы приведет уменьшение выпуска продукции.

Таким образом, можно сформулировать **критерии точки безубыточности**, при которой достигается максимальная прибыль: максимальная прибыль фирмы достигается при таком объеме производства, при котором предельный доход равен предельным издержкам.

## 2) Виды и составляющие издержек.

В рамках определения уровня безубыточности все затраты разделяют на две группы: **условно-переменные** и **условно-постоянные**. **FC (Fixed Cost)** – **постоянные (фиксированные) издержки** – денежные издержки, целом не изменяющиеся в зависимости от изменения объема 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.

Вместе с тем, при достижении определенного объема производства постоянные издержки могут быть увеличены на определенную величину.

**VC (Variable Cost)** – *переменные издержки* – издержки, меняющиеся пропорционально объему производства. Переменными издержками являются:

- затраты на сырье 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 дает возможность финансовому менеджеру:

- определить сроки окупаемости затрат;
- определить запас финансовой прочности предприятия;
- рассчитать оптимальную величину прибыли предприятия.

Кроме того, иногда выделяют и 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 объема производства, но не пропорционально его росту, а в другой пропорции. При конкретных расчетах 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.

**TC (Total Cost)** – *полные издержки* – сумма постоянных и переменных издержек. **Маржа на продажах С (Contribution margin)** – выручка за вычетом переменных издержек, в других терминах – средства, получаемые в распоряжение на операциях производства (закупки) и продажи партии товара. Причем эти средства не обязательно вкладывать на покрытие постоянных издержек. Предприятие будет рентабельным при условии, если маржинальный доход будет выше постоянных затрат.

**Маржинальные издержки MC (Marginal Cost)** – издержки производства на выпуск дополнительной продукции, при условии, что постоянные (фиксированные) издержки фирмой уже погашены.

**Невозвратные издержки SC (Sunk Cost)** – одноразовые невозвратные издержки.

**Упущенная выгода ОС (Opportunity Cost)** – доход, который можно было бы получить, но он не получен из-за отказа от использования предоставленной возможности.

В качестве примера в приложении 2 рассмотрен вид деятельности фирмы, специализирующейся на рынке прикладного программного обеспечения: от заказа на разработку программного продукта, – до его тиражирования.

### 3) Определение точки безубыточности.

Основным методом определения точки безубыточности является **CVP-анализ (Cost Value Profit - затраты, объем, прибыль)** [8], основанный на анализе соотношения затрат и объемов выпуска и прибыли.

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

$P = s \cdot x - (a + b \cdot x) = (s - b) \cdot x - a$ , (3.1), где  $P$  – прибыль фирмы;  $x$  – объем выпуска продукции;  $s$  – договорная цена продажи единицы продукции;  $a$  – величина фиксированных расходов;  $b$  – величина переменных издержек на единицу продукции.

В этом случае объем выпуска при котором достигается точка безубыточности (прибыль фирмы равняется нулю).

В некоторых случаях в качестве исходных данных при определении необходимого объема продаж используется величина предельного (маржинального) дохода от реализации единицы продукции как разницы между ценой и переменными издержками:  $m_0 = s_0 - b$ .

Следует отметить, что корректное использование методов CVP-анализа верно лишь **в ограниченном диапазоне объемов выпуска**. Ограниченностю проистекает, прежде всего, из-за того, что при достаточно большом объеме выпуска продукции перестают быть верными многие предпосылки, лежащие в основе CVP-анализа, например, неизменный характер и величина постоянных расходов, и т.д. В этой связи в литературе выделяют следующие условия применимости CVP-анализа при определении точки безубыточности:

- изменение общих затрат и выручки жестко определено и линейно в пределах области релевантности;
- 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.

В приведенном примере в число фиксированных расходов входит выплата ежемесячных отчислений и процентов по банковскому кредиту, который предположительно берется фирмой для реализации проекта на разработку программного продукта.

## 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ (СЕМИНАРСКИХ) ЗАНЯТИЙ

### **2.1. Практическое (семинарское) занятие №1 (2 часа).**

Основные понятия экономики информационной безопасности критически важных объектов.

### **2.2. Практическое (семинарское) занятие №2 (2 часа).**

Основные понятия менеджмента информационной безопасности критически важных объектов..

### **2.3. Практическое (семинарское) занятие №3 (2 часа).**

**Тема:** «Основы описания программного обеспечения как объекта продажи и поставки».

#### **2.3.1. Краткое описание проводимого занятия:**

Первым разделом экономической части ДП является *описание программного продукта*.

Программное обеспечение должно быть представлено как объект продажи и поставки и ориентировано на конкретного покупателя. При этом потенциальные покупатели могут сравнивать свои требования к программному продукту с возможностями, описанными в документе, проводить анализ конкурентных преимуществ с другими (аналогичными) программными системами.

Описание программного продукта должно соответствовать базовому Российскому стандарту в области информационных технологий ГОСТ Р 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].

Вышеуказанный документ должен содержать следующие основные разделы.

#### **Обозначения и указания:**

- обозначение описания продукта (описанию продукта присваивается индивидуальное обозначение - как документу, оно может иметь наименование, отличное от «описания продукта», например «Описание функциональных возможностей», «Информация о продукте», «Формуляр продукта», «Карта описания продукта»);
- обозначение продукта - по крайней мере должно включать наименование продукта и обозначение его версии или даты выпуска;
- поставщик (наименование и адрес по крайней мере одного поставщика);
- рабочая задача (возможности программного продукта, доступные конечному пользователю, а также предусмотренный результат функционирования рабочей системы.);
- 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).

#### **Карта описания программного продукта**

АИС «ТиС» - «Автоматизированная информационная система «Торговля и склад», версия 1.1.

Поставщик АИС «ТиС» – лаборатория объектно-ориентированного моделирования информационных систем кафедры автоматизации обработки информации ТУСУР. Адрес: 634045, г. Томск, ул. Вершинина, 74, корп. ФЭТ ТУСУРа, ауд. 405. тел. (3822) 41-44-70. АИС «ТиС» предназначена для информационной поддержки процессов управления складами и торговыми представительствами заказчика.

АИС «ТиС» соответствует ГОСТ Р 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. Данная СУБД может использоваться бесплатно.

Информацию о СУБД Firebird 1.5.1 можно получить на сайте производителя <http://www.ibphoenix.com>. АИС «ТиС» на стадии комплексных испытаний и опытной эксплуатации протестирована в работе с данной СУБД и является полностью совместимой с ней.

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 Государственный стандартом РФ ГОСТ Р ИСО/МЭК 15910—2002. Информационная технология. Процесс создания документации пользователя программного средства.

АИС «ТиС» снабжена программой инсталляции, обеспечивающей установку и настройку программного обеспечения в локальной сети (сервер) и на отдельные компьютеры (рабочие станции).

**2.4. Практическое (семинарское) занятие № 4 (2 часа).**

**Тема:** «Оценка качества программных средств по ГОСТ 28195-89.».

**2.4.1. Краткое описание проводимого занятия:**

**Надежность** - набор атрибутов, относящихся к способности программного средства сохранять свой уровень качества функционирования при установленных условиях за установленный период времени.

В описание продукта должна быть включена информация по процедурам сохранения данных. Данную информацию можно привести, указав, например, возможности резервирования данных с помощью функций операционной системы. Могут быть описаны дополнительные характеристики продукта, которые обеспечивают его

функциональные возможности (проверки достоверности исходных данных; защита против серьезных последствий ошибки пользователя; восстановление работоспособности программной системы при ошибках).

**Практичность** - набор атрибутов, относящихся к объему работ, требуемых для использования и индивидуальной оценки такого использования определенным или предполагаемым кругом пользователей, в том числе:

- интерфейс пользователя (тип интерфейса, например, строка команд, меню, окна, функциональная клавиша, функция подсказки);

- требуемая квалификация пользователя (конкретные знания, которые необходимо усвоить пользователю для применения соответствующего 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** - набор атрибутов, относящихся к соотношению между уровнем качества функционирования программного средства и объемом используемых ресурсов при установленных условиях (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; информация по эффективности применения продукта и удовлетворению им потребностей пользователя).

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

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

Пример

Исходные данные.

Тип системы: программно-информационная

Сложность системы: простая

Язык программирования – Delphi

Плановый срок разработки системы, установленный заказчиком – 12 месяцев.

Анализ объекта автоматизации

Декомпозириуем программную систему до уровня элементарных компонент и для оценки размеров каждой из них используем либо внешних экспертов, либо специалистов разработчика и заказчика.

В результате проведенного анализа получаем структур программы комплексов и компонент АИС «ТиС».

Для упрощения расчетов примем допущение, что работу по оцениванию системы проводит только один эксперт – представитель разработчика.

Эксперт должен дать оптимистическую (*o*), пессимистическую (*p*) и реалистическую (*b*) оценки строк исходного кода для каждой программной компоненты.

## **2.5. Практическое (семинарское) занятие № 5 (2 часа).**

**Тема:** «Декомпозиция системы защиты критически важных объектов для оценки каждого из компонент системы».

### **2.5.2. Краткое описание проводимого занятия:**

При декомпозиции целесообразно использовать следующие термины и определения:

- **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 бизнес-процессов организации;
- **программный комплекс** – совокупность программных компонент, реализующих конкретный 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.

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.

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

Средняя оценка по бета-распределению определяется путем умножения реалистической оценки на 4, добавлением оптимистической и пессимистической оценок и делением полученного результата на 6.

## **2.6. Практическое (семинарское) занятие № 6 (2 часа).**

**Тема:** «Определение трудозатрат, длительности реализации проекта и средней численности разработчиков для прямого метода».

### **2.6.2. Краткое описание проводимого занятия:**

Определение трудозатрат, длительности и средней численности специалистов на основе базовой конструктивной модели трудозатрат.

В основу **оценки трудозатрат** положена степенная функция следующего вида:

$$T = A \cdot R^{\alpha} \cdot E (KLOC) / 12, \quad (2.9)$$
 где  $T$  – трудозатраты, выраженные в человеко-месяцах;

$R$  (*KLOC*) – размерность программной системы, выраженная в тысячах строк кода.

Первый сомножитель  $A$  является доминирующим, он прямо пропорционален размерности программного обеспечения  $R$  и отражает линейную зависимость роста трудозатрат от размерности.

Второй сомножитель  $R^{\alpha} \cdot E$  отражает тот факт, что при увеличении размерности программной системы возрастает относительная трудоемкость разработки каждой строки программного кода за счет увеличения количества взаимосвязей между компонентами.

Значения параметров  $A$  и  $E$ , полученные путем статистической обработки данных по результатам реализации множества проектов, представлены в табл. 2.13 [4].

Оценки по СОСМО получены в результате обработки статистических данных по 160 реальным зарубежным проектам, а оценки по ПРОМЕТЕЮ – результат обобщения статистики – по 250 отечественным.

**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:

$$Z = T / D \quad (2.10)$$

Расчет длительности и, соответственно, численности специалистов для разработки программного обеспечения может быть произведен также из среднестатистической производительности труда программиста.

## 2.7. Практическое (семинарское) занятие № 7 (2 часа).

**Тема:** «Представление подсистемы защиты критически важных объектов в виде многоуровневой графической модели. Расчет количества функциональных точек».

### 2.7.2. Краткое описание проводимого занятия:

**Метод функциональных точек** (Function point – FP) основывается на том, что размеры программной системы оцениваются в терминах количества и сложности бизнес-процессов (функций), реализуемых в данном программном коде [2].

**Функциональная точка** – это комбинация свойств программного обеспечения:

- интенсивности использования ввода и вывода внешних данных;
- взаимодействия системы с пользователем;
- внешних интерфейсов;
- файлов, используемых системой.

Будущая система с использованием методологии структурного анализа и проектирования описывается в виде многоуровневой графической модели, представленной в виде совокупности взаимосвязанных функциональных диаграмм (пользовательских бизнес-процессов).

Каждый из бизнес-процессов включает в себя входные и выходные данные, преобразования и внешние интерфейсы.

Процедура оценивания размеров программной системы соотносится с одним из пользовательских бизнес-процессов и состоит из следующей последовательности этапов (ввод, вывод, опросы, структуры данных, интерфейсы):

- выделение множества бизнес-процессов;
- подсчет количества функциональных точек бизнес-процесса в разрезе каждой категории;
- определение весовых коэффициентов сложности каждой функции;
- учет факторов и требований среды разработки программной системы;
- вычисление интегральных показателей сложности;
- вычисление итогового количества функциональных точек;
- определение размеров программного комплекса в показателях LOC;
- 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;
- при декомпозиции сложной функции учитываются все логические преобразования с данными.

Расчет количества функциональных точек.

**Определение количества выводов.** Под выводами будем понимать следующие единицы информации, получаемые на выходе рассматриваемого бизнес-процесса:

- файлы, продуцируемые в данном бизнес-процессе для передачи другим бизнес-процессам, либо за пределы программной системы;
- единицы деловой информации, предназначенные для конечных пользователей, оформленные в виде экраных форм, либо бумажных документов.

Каждый из выводов, в зависимости от количества файлов, используемых при формировании выходов, рекомендуется отнести к одной из категорий сложности: простой, средний, сложный.

**Определение количества вводов.** Под вводами будем понимать следующие единицы информации, поступающие на вход рассматриваемого бизнес-процесса:

- 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

**Определение количества опросов ввода, вывода.** Под опросами будем понимать следующие действия, исполняемые программной системой в рассматриваемом бизнес-процессе:

- обращение к внешним процедурам, оформленным в виде специфических 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.

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.

**Определение количества файлов.** Под файлами будем понимать следующие единицы информации, использующиеся программной системой в рассматриваемом бизнес-процессе:

- внутренние логические файлы программной системы;
- структуры данных, представляющие собой первичную логическую группу пользовательских данных, которые постоянно находятся внутри границ программной системы;
- внешние файлы, доступные пользователям с помощью ввода, вывода, опросов, либо интерфейсов.

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. Каждый АРМ рассчитан на эксплуатацию конкретным подразделением организации.

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.

## **2.8. Практическое (семинарское) занятие № 8 (2 часа).**

**Тема:** «Определение трудозатрат, длительности и средней численности специалистов на основе базовой конструктивной модели трудозатрат при использовании метода функциональных точек».

### **2.8.2. Краткое описание проводимого занятия:**

В основу *оценки трудозатрат* положена степенная функция следующего вида:

$$T = A * R^{\lambda} E \text{ (KLOC)} / 12, \quad (2.9) \quad \text{где } T - \text{трудозатраты, выраженные в человеко-месяцах;}$$

$R$  (KLOC) – размерность программной системы, выраженная в тысячах строк кода.

Первый сомножитель  $A$  является доминирующим, он прямо пропорционален размерности программного обеспечения  $R$  и отражает линейную зависимость роста трудозатрат от размерности.

Второй сомножитель  $R^{\lambda} E$  отражает тот факт, что при увеличении размерности программной системы возрастает относительная трудоемкость разработки каждой строки программного кода за счет увеличения количества взаимосвязей между компонентами.

Значения параметров  $A$  и  $E$ , полученные путем статистической обработки данных по результатам реализации множества проектов, представлены в табл. 2.13 [4].

Оценки по СОСОМО получены в результате обработки статистических данных по 160 реальным зарубежным проектам, а оценки по ПРОМЕТЕЮ – результат обобщения статистики – по 250 отечественным.

**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:

$$Z = T / \bar{D} \quad (2.10)$$

Расчет длительности и, соответственно, численности специалистов для разработки программного обеспечения может быть произведен также из среднестатистической производительности труда программиста.

### **Метод функциональных точек:**

- 1) выделяется множество бизнес-процессов;
- 2) производится подсчет количества функциональных точек бизнес-процесса в разрезе каждой категории функций (простые/средние/сложные) для заданной последовательности этапов (ввод, вывод, опросы, структуры данных, интерфейсы) – таблицы 2.3-2.9, раздел 2.3.2 – формула 2.5;
- 3) производится учет факторов и требований среды разработки программной системы (табл. 2.10, 2.11) – формула 2.6;
- 4) по формуле 2.7. рассчитывается уточненное количество функциональных точек с учетом факторов внешней среды;

5) по формуле 2.8 определяется размерность программного обеспечения для конкретного языка программирования с учетом нормативов, представленных в таблице 2.12;

6) производится определение трудозатрат при заданной длительности и средней численности специалистов на основе базовой конструктивной модели трудозатрат – СОСМО: трудозатраты определяются по формуле 2.9, таблица 2.13;

7) по формуле 2.10 определяется средняя численность сотрудников, занятых в проекте.

## 2.9. Практическое (семинарское) занятие № 9 (2 часа).

**Тема:** «Определение размерности подсистемы защиты критически важных объектов по количеству объектов, атрибутов и их взаимосвязям в базе данных подсистемы защиты критически важных объектов.

### 2.9.2. Краткое описание проводимого занятия:

В результате анализа объекта автоматизации с помощью ER-моделирования строим концептуальную модель базы данных программной системы для определения количества таблиц (объектов) предметной области, связей и атрибутов (рис. 3).

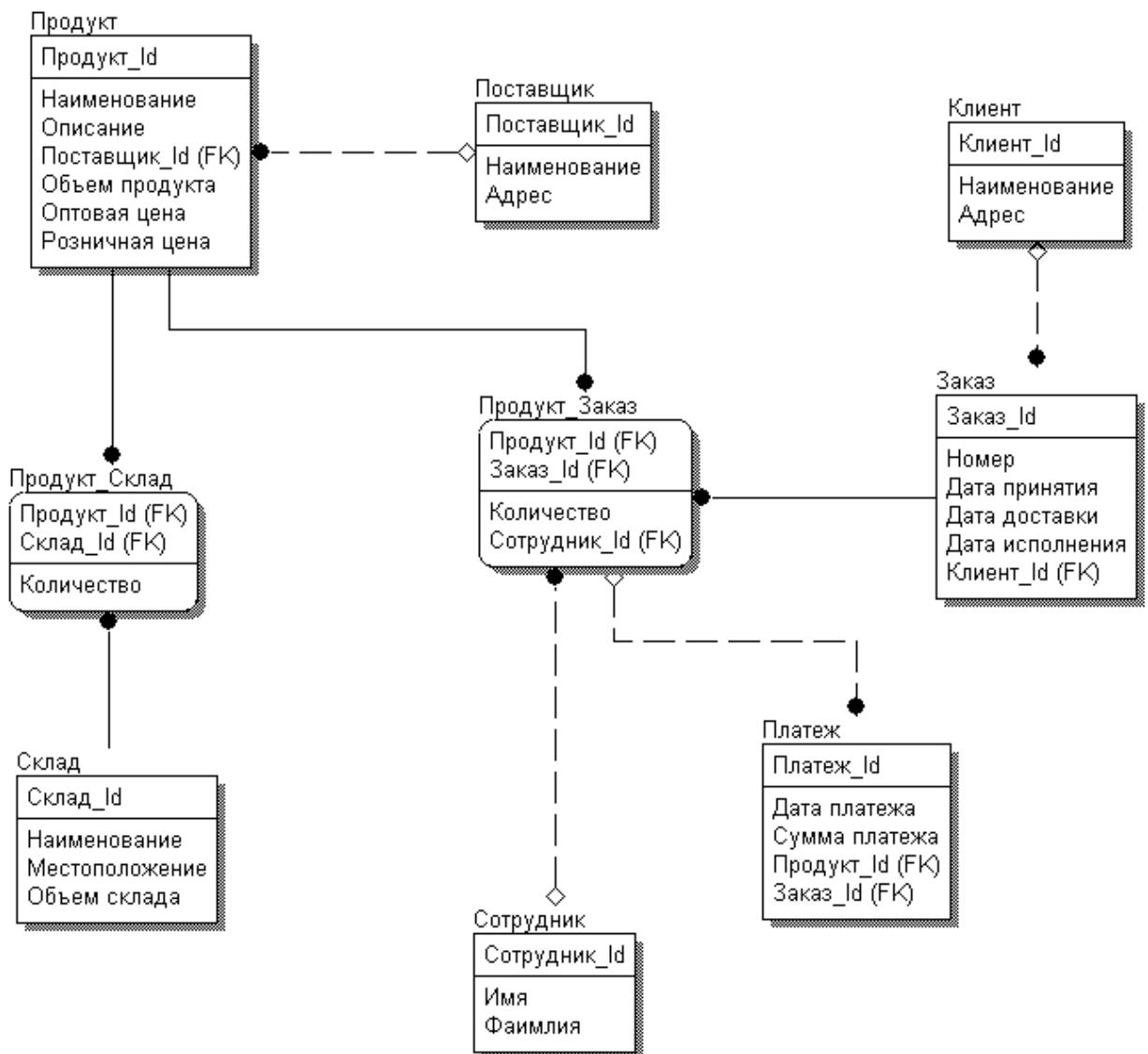


Рис. 3. Концептуальная модель базы данных

Анализируя построенную модель БД получаем:

$N$  – количество таблиц (объектов) = 9;

$1 K$  – количество взаимосвязей между объектами = 8;

$M$  – количество атрибутов на один объект =  $27/9 = 3$ .

Размерность программного обеспечения (в данном случае –базы данных) определяется по формуле 2.14:

$$R = 2N \cdot 5K \cdot 10M,$$

Подставляя в формулу результаты анализа, получаем размерность базы данных:

$$R = 2 \cdot 9 \cdot 5 \cdot 8 \cdot 10 \cdot 3 = 21600 \text{ полей БД}$$

Далее переходим к расчету ТЭП проекта и определим понятие «нормализованной величины» при создании программной системы. Этой величиной является количество формируемых атрибутов, входящих в электронные таблицы посредством установленных связей. При значениях  $N$ ,  $1 K$  и  $M$ , равных единице, величина, выражаяющая их количество равна 100.

Трудозатраты определяются по формуле 2.15 на основе статистических нормативов трудоемкости, приведенных в табл.

2.15:

$$T = 0.01 \cdot R \cdot \theta,$$

где  $\theta$  – норматив трудоемкости разработки программной системы, который характеризует собой категорию сложности разрабатываемой системы, в т.ч. размерность базы данных.

В нашем случае размерность базы данных (21600) находится в нормативном промежутке до 90 тыс. полей, что соответствует значению норматива  $\theta = 0,00566$ . Таким образом, трудоемкость будет равна:

$$T = 0.01 \cdot 21600 \cdot 0.00566 = 1.22 \text{ чел.-месяцев}$$

Длительность разработки, установленная заказчиком  $\Delta =$

12 месяцев, тогда средняя численность специалистов, которые должны быть привлечены к реализации программной системы составит:

$$Z = T / \Delta = 1.22 / 12 = 0.1 \text{ чел.}$$

Таким образом, применяя метод определения ТЭП на основе размерности базы данных ПС, мы определили следующие основные технико-экономические показатели разработки:

1) трудозатраты на разработку системы составят 1,22 человека-месяцев;

2) необходимые людские ресурсы при реализации системы = 0,1 чел.

### ВЫВОД:

При расчете технико-экономических показателей по трем методам, при заданной заказчиком длительности разработки 12 месяцев.

## 2.10. Практическое (семинарское) занятие № 10 (2 часа).

**Тема:** «Расчет трудозатрат на разработку базы данных большой размерности и длительности ее разработки.».

### 2.10.2. Краткое описание проводимого занятия:

Трудозатраты определяются по формуле 2.15 на основе статистических нормативов трудоемкости, приведенных в табл.

2.15:

$$T = 0.01 \cdot R \cdot \theta,$$

где  $\theta$  – норматив трудоемкости разработки программной системы, который характеризует собой категорию сложности разрабатываемой системы, в т.ч. размерность базы данных.

В нашем случае размерность базы данных (21600) находится в нормативном промежутке до 90 тыс. полей, что соответствует значению норматива  $\theta = 0,00566$ . Таким образом, трудоемкость будет равна:

$$T = 0.01 \cdot 21600 \cdot 0.00566 = 1.22 \text{чел.-месяцев}$$

Длительность разработки, установленная заказчиком  $\Delta = 12$  месяцев, тогда средняя численность специалистов, которые должны быть привлечены к реализации программной системы составит:

$$Z = T / \Delta = 1.22 / 12 = 0.1 \text{ чел}$$

Размерность программного обеспечения (в данном случае –базы данных) определяется по формуле 2.14:

$$R = 2N \cdot 5K_1 \cdot 10M,$$

Подставляя в формулу результаты анализа, получаем размерность базы данных:

$$R = 2 \cdot 9 \cdot 5 \cdot 8 \cdot 10 \cdot 3 = 21600 \text{ полей БД}$$

Далее переходим к расчету ТЭП проекта и определим понятие «нормализованной величины» при создании программной системы. Этой величиной является количество формируемых атрибутов, входящих в электронные таблицы посредством установленных связей. При значениях  $N$ ,  $1$   $K$  и  $M$ , равных единице, величина, выражающая их количество равна 100.

## 2.11. Практическое (семинарское) занятие № 11(2 часа).

**Тема:** «Расчет трудозатрат и длительности по основным этапам жизненного цикла создания подсистемы защиты критически важных объектов».

### 2.11.2. Краткое описание проводимого занятия:

Расчет трудозатрат и длительности по основным этапам жизненного цикла создания АСОИ.

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.

Используя эти распределения, по выражению 2.17 можно рассчитать среднюю численность сотрудников, занятых на каждом из этапов создания программной системы.

$$Z T \Delta_{ii} = \alpha / \beta, i=1,4 \quad (2.17)$$

## 2.12. Практическое (семинарское) занятие № 12 (2 часа).

**Тема:** «Определение численности каждого типа специалистов на каждом этапе жизненного цикла создания подсистемы защиты критически важных объектов. Расчет общего фонда заработанной платы на реализацию проекта подсистемы защиты критически важных объектов».

### 2.12.2. Краткое описание проводимого занятия:

Определение численности каждого типа специалистов на каждом этапе жизненного цикла создания АСОИ.

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

$Z = P \cdot Z_i, i=1,4, j=1,3, \quad (2.18)$  где  $P$  – относительная доля (%) специалистов  $j$ -го типа, привлекаемых для реализации проекта на  $i$ -ом этапе.

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 функциональных точек:

- трудоемкость ( $T$ ) = 15,7 чел.-месяцев;
- длительность ( $D$ ) = 12 месяцев.

Заполняем таблицу средней численности сотрудников, занятых на каждом из этапов создания ПС, используя статистические данные таблицы 2.16 и получаем расчетную таблицу 9.

По формуле 2.17 выполняем расчет средней численности сотрудников, занятых на каждом из этапов создания ПС.

$$Z_i = \alpha_i T / \beta_i D, \text{ где } i=1,4.$$

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

Следующий шаг – распределение специалистов по этапам жизненного цикла (ЖЦ) создания ПС, при этом численность каждого типа специалистов на каждом из этапов ЖЦ определяется по выражению 2.18 с использованием статистического распределения таблицы 2.17:

$Z_{ij} = P_{ij} \cdot Z_i, i=1,4 j=1,3$ , где  $P_{ij}$  – относительная доля (%) специалистов  $J$ -го типа, привлекаемых для реализации проекта на  $i$ -ом этапе.

Фонд заработной платы на проведение опытной эксплуатации определяется по формуле:  $S_{op} = Z_{op} \cdot t_{op} \cdot S_p \cdot 0,85$ , (2.22), где  $S_p$  – месячная базовая ставка программиста.

### 2.13. Практическое (семинарское) занятие № 13 (2 часа).

**Тема:** «Перечень работ на этапе опытной эксплуатации разрабатываемой подсистемы защиты критически важных объектов согласно ГОСТ 34.603-92. Нормативы трудоемкости опытной эксплуатации подсистемы защиты критически важных объектов».

#### 2.13.2. Краткое описание проводимого занятия:

Перечень работ на этапе опытной эксплуатации разрабатываемой АСОИ согласно ГОСТ 34.603-92.

На этапе опытной эксплуатации программной системы в соответствии с ГОСТ 34.603-92 производится заполнение нормативной базы, справочников, классификаторов, эксплуатация программного обеспечения в регламентном режиме, доработка программного обеспечения и рабочей документации в случае несоответствия текущей версии системы требованиям технического задания. Срок опытной эксплуатации оговаривается в техническом задании.

Численность сотрудников, привлекаемых к опытной эксплуатации определяется по формуле:

$Z_{op} = t_{op} \cdot N$ , (2.21) где  $t_{op}$  — срок опытной эксплуатации;  $N$  – норматив трудоемкости при проведении опытной эксплуатации.

Нормативы трудоемкости опытной эксплуатации АСОИ.

Нормативы трудоемкости опытной эксплуатации программной системы определяются исходя из среднестатистических нормативов.

Опытная эксплуатация проводится группой внедрения разработчика с привлечением, в случае необходимости, программистов, при этом относительный норматив заработной платы специалиста составляет 0,85.

Численность сотрудников, привлекаемых к опытной эксплуатации определяется по формуле 2.21:  $Z_{op} = t_{op} \cdot N$ , где  $t_{op}$  — срок опытной эксплуатации.

Установим срок опытной эксплуатации, согласованный с Заказчиком – 6 месяцев (при заданной длительности 12 месяцев).

Норматив трудоемкости при проведении опытной эксплуатации  $N$  определяется из таблицы 2.18 (категория сложности) – примем его равным 0,0095 чел.-месяцев (когда количество сеансов работы с системой в течение года (от 650 до 6000).

Численность сотрудников, привлекаемых для опытной эксплуатации составит:

$$Z_{оп} = 6 \cdot 0.0095 = 0.057(\text{чел.})$$

Фонд зарплаты сотрудников, привлекаемых для опытной эксплуатации, определяется по формуле 2.22:  $S_{оп} = Z_{оп} \cdot t_{оп} \cdot S_p \cdot 0.85$ , где  $S_p$  — месячная базовая ставка программиста.

В нашем случае вышеуказанный фонд составит:  $S_{оп} = 0.057 \cdot 6 \cdot 15000 \cdot 0.85 = 4360$  руб.

Общий фонд заработной платы на разработку и внедрение программной системы =  $230145 + 4360 = 234505$  руб.

## **2.14. Практическое (семинарское) занятие № 14 (2 часа).**

**Тема:** «Расчет фонда заработанной платы на проведение опытной эксплуатации разрабатываемой подсистемы защиты критически важных объектов».

### **2.14.2. Краткое описание проводимого занятия:**

Фонд заработной платы на проведение опытной эксплуатации определяется по формуле:  $S_{оп} = Z_{оп} \cdot t_{оп} \cdot S_p \cdot 0.85$ , (2.22), где  $S_p$  — месячная базовая ставка программиста.

Фонд зарплаты сотрудников, привлекаемых для опытной эксплуатации, определяется по формуле 2.22:  $S_{оп} = Z_{оп} \cdot t_{оп} \cdot S_p \cdot 0.85$ , где  $S_p$  — месячная базовая ставка программиста.

В нашем случае вышеуказанный фонд составит:  $S_{оп} = 0.057 \cdot 6 \cdot 15000 \cdot 0.85 = 4360$  руб.

Общий фонд заработной платы на разработку и внедрение программной системы =  $230145 + 4360 = 234505$  руб.

## **2.15. Практическое (семинарское) занятие № 15 (2 часа).**

**Тема:** «Составление сметы затрат на разработку и внедрение подсистемы защиты критически важных объектов».

### **2.15.2. Краткое описание проводимого занятия:**

Структура договорной цены

1. Наименование статей расходов:
  - 1) Оплата труда непосредственных исполнителей.
  - 2) Страховые взносы в ПФР, ФСС, Федеральный ФОМС.
  - 3) Увеличение стоимости основных средств.
  - 4) Амортизация средств вычислительной техники стоимостью от 40000 рублей и выше.
  - 5) Прочие выплаты (командировки в части суточных).
  - 6) Транспортные услуги (командировки в части оплаты транспортных расходов).
  - 7) Прочие услуги (командировки в части проживания, оплата услуг сторонних организаций по НИР, ОКР и технологическим работам, договора подряда с ЕЧН).
  - 8) Прочие расходы (расходные материалы, услуги связи и т.д.).
  - 9) Накладные расходы организации разработчика (расходы на АУП, охрану, обслуживающий персонал и т.д.).
  - 10) Фонд развития производства.
  - 11) Налог на добавленную стоимость.

Договорная цена на разработку и внедрение программной системы имеет, в основном, типовую структуру, которая включает в себя соответствующие статьи расходов, приведенные и подробно изложенные в разделе 2.4.3.

Основополагающим элементом, из которого будет произведен расчет стоимости проекта, является рассчитанный выше общий фонд заработной платы (234505 руб.).

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

Стоимость персонального компьютера, приобретенного для выполнения проекта, составляет 50000 рублей, при этом амортизационные отчисления для средств ВТ стоимостью свыше 40000 рублей, согласно действующему законодательству будут производиться в течение установленного в организации срока полезного использования техники (в нашем случае – 36 месяцев), а за 12 месяцев, определенных Заказчиком на разработку системы, составят  $50000/36*12 = 16667$  рублей.

Кроме того, должны быть предусмотрены расходы на лицензионное программное обеспечение, если его приобретение необходимо для разработки и эксплуатации программного продукта.