

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Санкт-Петербургский государственный университет телекоммуникаций**  
**им. проф. М. А. Бонч-Бруевича»**

Факультет «Информационных систем и технологий»  
Кафедра «Информационных управляющих систем»

Дисциплина: **Гипертехнологии информационных инфраструктур**

**ПРАКТИЧЕСКАЯ РАБОТА**  
**«Оценка трудоемкости разработки программных средств**  
**информационной системы»**

Выполнил: студент \_\_\_\_\_ (\_\_\_\_\_)  
группа

Проверил: \_\_\_\_\_ (проф. Мошак Н.Н.)

2017г.

ГОСТ Р ИСО/МЭК 12207-99	Информационная технология. Процессы жизненного цикла программных средств.
ГОСТ Р ИСО/МЭК 14764–2002	Информационная технология. Сопровождение программных средств.
ГОСТ 19.101–77	Виды программ и программных документов.
ГОСТ 19781–90	Обеспечение систем обработки информации программное. Термины и определения.
ГОСТ 28806–90	Качество программных средств. Термины и определения.
ГОСТ 34.003–90	Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
ГОСТ 34.601–90	Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
ГОСТ 20886–85	Организация данных в системах обработки данных. Термины и определения.

## **1. ЦЕЛЬ РАБОТЫ И ЗАДАНИЕ**

Целью курсового проекта является расчет трудоемкости разработки программных средств информационной системы, затрат времени и численности программистов по предлагаемой методике (далее - Методика) с использованием каталога функций с учетом требований **ТЕХНИЧЕСКОГО ЗАДАНИЯ**

### **Техническое задания на разработку ПС**

#### **1.1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**

##### **1.1.1. Назначение работы**

Программное средство мониторинга финансовых операций, выполняемых через банкомат, предназначено для автоматизации:

- учета выданных карт клиентам банка;
- учета операций по снятию наличных в банкоматах Банка;
- учета операций по оплате услуг оператора сотовой связи через банкоматы;
- составления отчета о выполненных операциях;
- анализа выполненных операций для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения.

Программное средство должно обеспечить возможность:

- автоматизированной обработки информации о проведенных операциях и состоянии личного счета клиентов Банка, использующих возможность выполнения финансовых операций по снятию наличных и оплате услуг оператора сотовой связи через банкоматы Банка;
- составления отчета по выполненным финансовым операциям и состоянию счета клиентов;
- анализа информации, полученной от банкоматов и планирования их обслуживания.

##### **1.1.2. Область применения**

Программное средство, разрабатываемое в соответствии с данным техническим заданием, предназначено для применения в подразделениях Банка России, осуществляющих контроль финансовых операций. ПС создается с целью автоматизации учета финансовых операций, выполняемых клиентами Банка с использованием кредитных карт.

#### **1.2. ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ**

##### **1.2.1. Требования к структуре ПС**

Необходимая для функционирования программного средства информация поступает на вход ПС из следующих источников:

- банкомат (передает информацию о выполнении финансовых операций);
- оператор ПС (выполняет ввод, получение и удаление информации о клиентах Банка, которым были выданы кредитные карты);
- администратор ПС (выполняет ввод, получение и удаление информации о новых банкоматах и операторах сотовой связи, вводе данных о пополнении денежных средств банкоматов).

В результате работы программное средство формирует:

- отчет о выполненных операциях;
- результаты анализа наличия денежных средств в банкоматах и примерные сроки их пополнения.

Каждая функция программного средства предполагает возврат сообщения об ошибках.

Программное средство должно состоять из следующих компонентов:

- «АРМ Оператора», предназначенного для работы оператора с программным средством;
- «АРМ Администратора», предназначенного для работы администратора с программным средством;
- «База Данных», предназначенного для хранения информации о выданных кредитных картах, операций по снятию наличных, оплаты услуг сотовой связи;
- «Отчет», предназначенного для составления отчета о выполняемых операциях.
- «Анализ», предназначенного для прогнозирования сроков пополнения и количества необходимых наличных денег для банкоматов.

Основными источниками данных для программного средства являются:

- база данных Банка, содержащая информацию о выданных клиентам кредитных картах и проводимых по ним операциям;
- данные, получаемые от банкоматов Банка.

При этом необходимо отметить:

– Компоненты «АРМ оператора» и «АРМ администратора» реализуют минимально необходимый интерфейс. Компонент «АРМ администратора» не реализует функций по получению и удалению информации о банкоматах и операторах сотовой связи, а реализует только ввод этих данных. Для макета также не обязательна полноценная реализация функций обеспечения безопасности информации, хотя в минимальном варианте данная возможность должна присутствовать. Таким образом, программное средство реализует следующие функции:

**1) Компонент «АРМ оператора»:**

- добавление, получение и удаление информации о клиенте Банка в базе данных;
- выполнение запросов к базе данных для получения состояния счета, снятия наличных средств со счета, перевода денежных средств клиента со счета в Банке при оплате услуг или перевода денежных средств на счет клиента Банка;

**2) Компонент «АРМ администратора»:**

- добавление информации о банкомате, операторе сотовой связи;
- выполнение запросов к базе данных для ввода информации о пополнении денежных средств банкоматов.

**3) Компонент «База данных»:**

- формирование структуры базы данных;
- обеспечение безопасности передачи данных при работе с БД.

**4) Компонент «Анализ»:**

- проведение анализа о выполненных операциях для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения;
- формирование отчета по анализу.

**3) Компонент «Отчет»:**

- формирование отчета о выполненных операциях.

### **1.2.2. Требования к функциональным возможностям ПС**

Программное средство должно обеспечивать выполнение следующих функций:

- добавление оператором ПС информации о клиенте Банка в базу данных при получении клиентом кредитной карты;
- получение/удаление оператором ПС информации о клиенте Банка из базы данных;
- добавление администратором ПС информации о новом банкомате;
- получение/ удаление администратором ПС информации о банкомате;
- добавление администратором ПС информации об операторе сотовой связи;
- получение/ удаление администратором ПС информации об операторе сотовой связи;

- выполнении администратором ПС операции ввода информации о пополнении денежных средств банкоматов;
- выполнение оператором ПС или клиентом Банка операции получения состояния счета Клиента Банка;
- выполнение оператором ПС или клиентом Банка операции снятия наличных средств со счета клиента Банка;
- выполнение оператором ПС или клиентом Банка операции перевода денежных средств клиента со счета в Банке при оплате услуг;
- выполнение оператором ПС операции перевода денежных средств на счет клиента Банка;
- формирование оператором ПС отчета о выполненных операциях;
- получение администратором ПС анализа о выполненных операциях для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения.

### **1.2.3. Технические и системные требования к ПС**

Программное средство мониторинга финансовых операций, выполняемых через банкомат, должно функционировать на ТС со следующими минимальными требованиями по аппаратному и программному обеспечению:

- CPU P3, RAM 128 MB;
- 50 MB дискового пространства;
- ОС Windows XP;
- СУБД SQL Server 6.0.

### **1.2.4. Требования к надежности**

Программное средство мониторинга финансовых операций, выполняемых через банкомат, должно обладать надежностью, обеспечивающей работу ПС в рабочем режиме и оперативное восстановление работоспособности при сбоях. ПС должно иметь необходимый высокий уровень защиты информации в БД и информации, поступающей в ПС от банкоматов Банка.

В ПС должны быть предусмотрены:

- контроль за целостностью данных на уровне СУБД;
- сохранение целостности данных в БД при нештатном завершении программы;
- сохранение работоспособности ПС при некорректных действиях администратора или оператора ПС.

### **1.2.5. Требования к эргономике**

Программное средство должно использовать средства построения графического интерфейса пользователя, обеспечиваемых операционной системой Windows.

Сообщения, выдаваемые программным средством, должны выдаваться на русском языке за исключением стандартных системных сообщений.

## **1.3. ТРЕБОВАНИЕ К ДОКУМЕНТИРОВАНИЮ**

1.3.1. Состав и содержание документации должны соответствовать требованиям ГОСТ 19.101–77.

3.2. Перечень документов должен включать:

- 1) Техническое задание на разработку ПС.
- 2) Пояснительная записка;
- 3) Описание информационного обеспечения программы;
- 4) Спецификация;
- 5) Формуляр;
- 6) Программа и методика предварительных испытаний;

- 7) Программа и методика приемочных испытаний;
- 8) Руководство пользователя.

## **2. МЕТОДИКА ОЦЕНКИ ТРУДОЕМКОСТИ РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ С ИСПОЛЬЗОВАНИЕМ КАТАЛОГА ФУНКЦИЙ**

### **2.1. Общие положения**

2.1.1. *Процесс разработки* ПС состоит из работ, выполняемых разработчиком, и включает работы по анализу требований к ПС, проектированию, программированию, тестированию и вводу в действие ПС. В соответствии с этим перечнем работ процесс разработки ПС в Методике состоит из следующих подпроцессов:

- *анализ требований к ПС;*
- *проектирование;*
- *программирование;*
- *тестирование;*
- *ввод в действие ПС.*

#### 1) Анализ требований к ПС:

- подготовка процесса;
  - определение или выбор модели жизненного цикла ПС;
  - документальное оформление выходных результатов в соответствии с процессом документирования;
  - выполнение вспомогательных процессов;
  - выбор стандартов, методов, инструментария, языков программирования;
  - разработка плана проведения процесса разработки.
- анализ требований:
  - анализ области применения разрабатываемого ПС с точки зрения определения требований к нему. Технические требования к ПС должны охватывать: функции и возможности ПС; организационные требования; требования пользователя; требования безопасности и защиты; эргономические требования; требования к интерфейсам; эксплуатационные требования; требования к сопровождению и квалификационные требования. Технические требования к ПС должны быть документально оформлены;
  - оценка и документальное оформление оценки требований к ПС с учетом учета потребностей заказчика, соответствия потребностям заказчика, тестируемости, выполнимости проектирования архитектуры ПС, возможности эксплуатации и сопровождения.

#### 2) Проектирование:

- проектирование программной архитектуры:
  - трансформирование требований к ПС в архитектуру, которая описывает общую структуру ПС и определяет компоненты ПС. Распределение требований к ПС между его компонентами. Документальное оформление архитектуры ПС;

- разработка и документальное оформление общего (эскизного) проекта внешних интерфейсов и интерфейсов между компонентами ПС;
  - разработка и документальное оформление общего (эскизного) проекта базы данных;
  - разработка и документальное оформление предварительной версии эксплуатационной документации;
  - разработка и документальное оформление предварительных требований к тестированию ПС, разработка графика сборки ПС;
  - оценка и документальное оформление архитектуры ПС и эскизных проектов.
- техническое проектирование ПС:
- разработка и документальное оформление технического проекта для ПС. Компоненты ПС должны быть уточнены на уровне программных модулей, которые можно программировать, компилировать и тестировать независимо. Распределение технических требований к компонентам между программными модулями;
  - разработка технического проекта внешних интерфейсов, интерфейсов между программными модулями;
  - разработка технического проекта базы данных;
  - уточнение эксплуатационной документации;
  - определение и документальное оформление требований к испытаниям и программе испытаний программных модулей;
  - оценка технического проекта и требований к тестированию, документальное оформление оценки.

### 3) Программирование:

- программирование и тестирование компонентов ПС:
- разработка и документальное оформление каждого программного модуля и базы данных;
  - разработка программы, методики испытаний и данных для тестирования каждого программного модуля и базы данных;
  - тестирование каждого программного модуля и базы данных;
  - уточнение эксплуатационной документации;
  - уточнение программы сборки ПС;
  - оценка запрограммированных элементов ПС и документальное оформление оценки.
- сборка ПС:
- разработка плана сборки и тестирования, документальное оформление плана;
  - сборка и тестирование программных модулей, документальное оформление результатов;
  - разработка и документальное оформление наборов тестов, контрольных примеров, процедур испытаний;
  - оценка и документальное оформление оценки плана сборки, проекта, запрограммированного ПС, проведенных испытаний, результатов тестирования, эксплуатационной документации.

### 4) Тестирование ПС:

- проведение тестирования на соответствие требованиям к ПС;
- анализ и документальное оформление результатов тестирования;
- уточнение эксплуатационной документации (при необходимости);

- доработка ПС по результатам тестирования (при необходимости).
- 5) Ввод в действие ПС:
  - ввод в действие ПС:
    - разработка и документальное оформление работ по вводу в действие ПС в среде эксплуатации (проведение предварительных испытаний, опытная эксплуатация);
    - ввод в действие ПС в соответствии с планом. Документальное оформление работ.
  - обеспечение приемки ПС:
    - обеспечение проведения заказчиком оценки готовности к приемке и приемочным испытаниям. Документальное оформление результатов оценки готовности;
    - поставка ПС заказчику;
    - первоначальное и непрерывное обучение и поддержка персонала заказчика.

2.1.2. Для расчета трудоемостей подпроцессов разработки ПС, а также базовой трудоемкости разработки ПС, в Методике использованы мультипликативные модели следующего вида:

$$T = A \cdot F_{кол_1}^{B_1} \cdot \dots \cdot F_{кол_q}^{B_q} \cdot F_{кат_1} \cdot \dots \cdot F_{кат_g}, \quad (2.1)$$

где  $T$  – рассчитываемая трудоемкость;

$F_{кол_1}, \dots, F_{кол_q}$  – количественные переменные;

$F_{кат_1}, \dots, F_{кат_g}$  – категориальные переменные с дискретными уровнями;

$A, B_1, \dots, B_q$  – числовые параметры.

В качестве количественных переменных в моделях используются следующие характеристики:

- объем ПС (строк исходного текста);
- объем документации (строк текста документации);
- объем доработок кода (строк исходного текста);
- объем доработок документации (строк);
- количество установленных экземпляров ПС;
- число пользователей.

В качестве категориальных переменных выступают поправочные коэффициенты, учитывающие влияние факторов на трудоемкость разработки ПС. Трудоемкость разработки ПС рассчитывается с учетом следующих факторов:

- сложность ПС;
- степень новизны ПС;
- уровень требований к показателям качества ПС (определяются заказчиком ПС);
- условия и средства разработки ПС;
- опыт и квалификация разработчика.

Перечень поправочных коэффициентов, учитывающих влияние вышеперечисленных факторов, приведен в таблице 2.1.

Перечень поправочных коэффициентов, учитываемых при расчете трудоемкости разработки ПС, в разрезе учитываемых факторов

Таблица 2.1

Факторы, учитываемые при расчете трудоемкости разработки ПС	Поправочные коэффициенты, учитывающие влияние фактора
Сложность ПС	Коэффициент, учитывающий сложность функции
Степень новизны ПС	Коэффициент, учитывающий степень новизны ПС
Уровень требований к показателям качества ПС	Коэффициент, учитывающий требования к надежности ПС
	Коэффициент, учитывающий требования к производительности ПС
	Коэффициент, учитывающий требования к уровню информативности документации на фазах жизненного цикла ПС
Условия и средства разработки ПС	Коэффициент, учитывающий используемые средства проектирования
	Коэффициент, учитывающий конкретные условия и средства разработки
	Коэффициент, учитывающий используемые технологии разработки
	Коэффициент, учитывающий используемую среду разработки
	Коэффициент, учитывающий используемые средства тестирования
	Коэффициент, учитывающий размер базы данных
Опыт и квалификация разработчика	Коэффициент, учитывающий уровень квалификации аналитиков
	Коэффициент, учитывающий опыт работы аналитиков в данной предметной области
	Коэффициент, учитывающий уровень квалификации проектировщиков
	Коэффициент, учитывающий опыт работы проектировщиков с используемыми средствами проектирования
	Коэффициент, учитывающий опыт работы программистов с данным средством разработки
	Коэффициент, учитывающий опыт разработки программных средств подобного типа
	Коэффициент, учитывающий уровень квалификации программистов
	Коэффициент, учитывающий уровень квалификации персонала, осуществляющего тестирование
	Коэффициент, учитывающий уровень квалификации персонала, осуществляющего ввод в действие ПС

При расчете трудоемкости разработки ПС в Методике дополнительно используются следующие поправочные коэффициенты:

- коэффициент, учитывающий степень повторного использования функции;
- коэффициент, учитывающий влияние сроков работ на трудоемкость.

2.1.3. **Поправочные коэффициенты**, используемые при расчете трудоемкости разработки ПС, разделяются на три уровня влияния:

- уровень расчета;
- уровень подпроцесса разработки;
- уровень функции.

Коэффициенты уровня расчета оказывают влияние на трудоемкость разработки программного средства в целом.

Коэффициенты уровня подпроцесса разработки оказывают влияние на трудоемкость отдельных подпроцессов разработки ПС.

Коэффициенты уровня функции оказывают влияние на трудоемкость разработки отдельной функции программного средства.

Перечень поправочных коэффициентов в разрезе уровней влияния приведен в таблице 2.2.

Перечень поправочных коэффициентов, учитываемых при расчете трудоемкости разработки ПС, в разрезе уровней влияния

Таблица 2.2

Уровень влияния	Поправочные коэффициенты, относящиеся к уровню влияния
Уровень расчета	Коэффициент, учитывающий степень новизны ПС
	Коэффициент, учитывающий требования к надежности ПС
	Коэффициент, учитывающий требования к производительности ПС
	Коэффициент, учитывающий требования к уровню информативности документации на фазах жизненного цикла ПС
	Коэффициент, учитывающий используемые технологии разработки
	Коэффициент, учитывающий опыт разработки программных средств подобного типа
	Коэффициент, учитывающий влияние сроков работ на трудоемкость
Уровень подпроцесса разработки	Коэффициент, учитывающий уровень квалификации аналитиков
	Коэффициент, учитывающий опыт работы аналитиков в данной предметной области
	Коэффициент, учитывающий уровень квалификации проектировщиков
	Коэффициент, учитывающий опыт работы проектировщиков с используемыми средствами проектирования
	Коэффициент, учитывающий используемые средства проектирования
	Коэффициент, учитывающий уровень квалификации программистов
	Коэффициент, учитывающий используемую среду разработки
	Коэффициент, учитывающий уровень квалификации персонала, осуществляющего тестирование
	Коэффициент, учитывающий используемые средства тестирования
	Коэффициент, учитывающий размер базы данных
	Коэффициент, учитывающий уровень квалификации персонала, осуществляющего ввод в действие ПС
Уровень функции	Коэффициент, учитывающий сложность функции
	Коэффициент, учитывающий степень повторного использования функции
	Коэффициент, учитывающий конкретные условия и средства разработки
	Коэффициент, учитывающий опыт работы программистов с данным средством разработки

Поправочные коэффициенты используются для уточнения значения трудоемкости разработки ПС. Поправочный коэффициент представляет собой дискретную величину. Значение поправочного коэффициента выбирается из соответствующих таблиц Методики. Таблицы значений поправочных коэффициентов для расчета трудоемкости разработки ПС приведены в приложениях 2–4 Методики.

Если для определения уровня какого-либо поправочного коэффициента отсутствуют необходимые данные, то следует принять значение этого коэффициента равным единице и таким образом, не учитывать его влияние при расчете трудоемкости.

### 3.1. Порядок расчета трудоемкости разработки ПС

3.1.1. Для расчета трудоемкости разработки ПС необходимо определить состав функций и примерный объем (размер) этого программного средства с учетом объема выбранных функций. Состав функций ПС определяется на основании имеющейся в техническом задании информации о составе функций разрабатываемого ПС. Объем ПС определяется исходя из выполняемых им функций путем суммирования объемов этих функций.

Современные программные средства разрабатываются как набор функционально взаимосвязанных структурных единиц, поэтому введение понятия «компонент» упрощает логическое структурирование расчетов. Функции, включаемые в состав программного средства, могут группироваться в компоненты.

Компонент выступает как средство разделения функций, входящих в функционально и логически обособленные части ПС. Выделение компонентов целесообразно в том случае, когда компонент является частью проектного решения и служит для логического структурирования программного средства.

Объем каждой отдельной функции разрабатываемого ПС ( $V_{i_j}$ ) выражается числом строк исходного текста, написанных непосредственно разработчиком на используемом средстве разработки (без учета текста, сгенерированного автоматически, использованного из библиотек и т.д.). **Для определения объема каждой отдельной функции используется Каталог функций программных средств (Приложение 1 Методики).**

Функции, приведенные в Каталоге функций, сгруппированы по типам и классам функций. Классы всегда определяются внутри типов функций.

Учитывая тот факт, что конкретная функция, реализованная с использованием различных средств разработки, может иметь различный объем (число строк исходного текста) в Методике в качестве единицы объема введено понятие «строка исходного текста на условном языке». Значения объемов функций, приведенных в Каталоге функций, выражены в строках исходного текста на условном языке.

Каталог содержит функции различной степени детализации; некоторые из них могут пересекаться или вкладываться друг в друга. При использовании Каталога функций необходимо выбирать из него наиболее детальные функции, чтобы повысить точность расчета итоговой трудоемкости.

Объемы функций в Каталоге представлены как в виде одиночного значения, которое непосредственно используется в расчете, так и в виде диапазона значений. В случае если используется объем функции с диапазоном значений, возможны два способа определения объемов функций разрабатываемого ПС:

**1-й способ.** Этот способ применяется, когда разработчик затрудняется экспертным методом оценить объем функции. При этом выбирается **среднее** значение объема соответствующей функции ( $V_i$ ), выраженное числом строк исходного текста на условном языке из диапазона, указанного в таблице 1.1 Приложения 1.

2-й способ. Этот способ применяется, когда разработчик может оценить объем функции на используемом языке разработки. Значение объема функции ( $Vu_i$ ) выбирается **экспертным методом** из допустимого диапазона значений. Для этого границы объема данной функции (строк исходного текста, написанного на условном языке), приведенные в таблице 1.1 Приложения 1, пересчитываются для конкретного языка (средства) разработки с помощью соответствующего коэффициента (таблица 1.2 Приложения 1) по следующим формулам:

$$\begin{aligned} Vmn_i &= Vn_i / kp_i, \\ Vmv_i &= Vv_i / kp_i, \end{aligned} \quad (3.1)$$

где  $Vmn_i$  – нижняя граница объема  $i$ -й функции ПС на используемом языке (средстве) разработки;

$Vmv_i$  – верхняя граница объема  $i$ -й функции ПС на используемом языке (средстве) разработки;

$Vn_i$  – нижняя граница объема  $i$ -й функции ПС на условном языке (таблица 1.1 Приложения 1);

$Vv_i$  – верхняя граница объема  $i$ -й функции ПС на условном языке (таблица 1.1 Приложения 1);

$kp_i$  – переводной коэффициент, определяемый в зависимости от используемого средства разработки (таблица 1.2 Приложения 1);

$i = 1 \div n$  – количество функций ПС.

Если значение объема функции на языке разработки (например, в случае расчета по написанному коду ПС) находится ниже или выше рекомендуемого диапазона, то в качестве значения объема функции ( $Vu_i$ ) следует принимать нижнюю ( $Vmn_i$ ) или верхнюю ( $Vmv_i$ ) границу допустимого диапазона значений соответственно.

3.1.2 При использовании 2-го способа определения объема функции значение объема функции ( $V_i$ ), выраженное числом строк исходного текста на условном языке, определяется исходя из объема функции ( $Vu_i$ ) на используемом языке (средстве) разработки по формуле:

$$V_i = Vu_i \cdot kp_i, \quad (3.2)$$

где  $Vu_i$  – объем  $i$ -й функции ПС на используемом языке (средстве) разработки;

$kp_i$  – переводной коэффициент, определяемый в зависимости от используемого средства разработки (таблица 1.2 Приложения 1);

$i = 1 \div n$  – количество функций ПС.

3.1.3. Отдельные функции разрабатываемого программного средства могут реализовываться неоднократно в неизменном виде или с некоторыми изменениями. Простое суммирование объемов одинаковых функций может привести к завышению трудоемкости их разработки.

Скорректированный объем  $i$ -ой функции ПС ( $V_{\hat{m}_i}$ ) определяется по формуле:

$$V_{\hat{m}_i} = V_i \cdot r_i \cdot k_i, \quad (3.3)$$

где  $V_i$  – объем  $i$ -ой функции ПС (определяемый на основании пункта 0 либо непосредственно по Каталогу функций – таблица 1.1 Приложения 1);

$r_i$  – число реализаций  $i$ -ой функции;

$k_i$  – коэффициент, учитывающий степень повторного использования  $i$ -ой функции (может принимать значения от 0 до 1).

Под степенью повторного использования функции подразумевается часть программного кода функции, которая пишется разработчиками заново.

$i = 1 \div n$  – количество функций ПС.

3.1.4. Скорректированный объем функции ПС ( $V_{mi}$ ) не учитывает влияние различных субъективных и объективных факторов на трудоемкость разработки ПС. Для учета этого влияния используются поправочные коэффициенты уровня функции (Приложение 2). Сложность функции определяется на основе характеристик сложности соответствующего типа операций.

Для этого каждый класс функций, приведенный в Каталогах функций, отнесен к одному из следующих типов операций:

- управляющие операции;
- вычислительные операции;
- операции, зависящие от аппаратуры;
- операции управления данными;
- операции управления пользовательского интерфейса.

Уровни сложности операций ( $L_v$ ) каждого класса функций определяются по одной из соответствующих таблиц 2.1–2.5 Приложения 2. В зависимости от уровня сложности ( $L_v$ ) определяется соответствующее значение коэффициента сложности функции ( $K_{сложн}$ ) по таблице 2.6 Приложения 2.

Скорректированный с учетом влияния различных факторов объем функции ПС ( $Vk_i$ ) определяется по формуле:

$$Vk_i = Vm_i \cdot K_{сложн} \cdot K_{ср.разр} \cdot K_{опыт} \quad (3.4)$$

где  $Vm_i$  – скорректированный объем  $i$ -ой функции ПС, определяемый в соответствии с пунктом 0;

$K_{сложн}$  – коэффициент сложности функции (таблица 2.6. Приложения 2);

$K_{ср.разр}$  – коэффициент, учитывающий конкретные условия и средства разработки ПС (таблица 2.7 Приложения 2);

$K_{опыт}$  – коэффициент, учитывающий опыт работы программистов с данным средством разработки (таблица 2.8 Приложения 2).

3.1.5. Объем ПС ( $V$ ) рассчитывается как сумма объемов функций, входящих в состав ПС, и определяется по формуле:

$$V = \sum_{j=1}^n Vk_j, \quad (3.5)$$

где  $Vk_j$  – объем  $j$ -ой функции, определяемый в соответствии с пунктом 0;

$j = 1 \div n$  – количество функций ПС.

3.1.6. Исходя из объема ПС ( $V$ ) определяется базовая трудоемкость разработки ПС ( $T_{баз}$ ). При расчете базовой трудоемкости разработки ПС учитывается влияние следующих факторов:

- степень новизны ПС;
- уровень требований к показателям качества ПС;
- условия и средства разработки ПС;
- опыт и квалификация разработчика.

Для учета влияния вышеперечисленных факторов используются поправочные коэффициенты уровня расчета (Приложение 3).

Базовая трудоемкость разработки ПС ( $T_{баз}$ ) с учетом влияния этих факторов определяется по формуле:

$$T_{баз} = A \cdot V^C \cdot K_n \cdot K_{над} \cdot K_{произв} \cdot K_{докум} \cdot K_{тех} \cdot K_{оп} , \quad (3.6)$$

где  $V$  – объем ПС, определяемый в соответствии с пунктом 0;  
 $K_n$  – коэффициент, учитывающий степень новизны ПС (таблица 3.1 Приложения 3);  
 $K_{над}$  – коэффициент, учитывающий требования к надежности ПС (таблица 3.2 Приложения 3);  
 $K_{произв}$  – коэффициент, учитывающий требования к производительности ПС (таблица 3.3 Приложения 3);  
 $K_{докум}$  – коэффициент, учитывающий требования к уровню информативности документации на фазах жизненного цикла ПС (таблица 3.4 Приложения 3);  
 $K_{тех}$  – коэффициент, учитывающий использование технологий разработки (см. пункт 0);  
 $K_{оп}$  – коэффициент, учитывающий опыт разработки программных средств подобного типа (таблица 3.5 Приложения 3);  
 $A, C$  – поправочные коэффициенты, рассчитанные на основе собранной статистики ( $A = 0,19, C = 0,74$ ).

3.1.7. Значение коэффициента, учитывающего используемые технологии разработки ( $K_{тех}$ ), определяется исходя из структуры разрабатываемого ПС и технологии взаимодействия компонентов ПС. Значение коэффициента  $K_{тех}$  определяется по формуле:

$$K_{тех} = K_{сmp} + \sum_{i=1}^l K_{mi} , \quad (3.7)$$

где  $K_{сmp}$  – коэффициент, учитывающий структуру разрабатываемого программного средства (таблица 3.6 Приложения 3);  
 $K_{mi}$  – коэффициент, учитывающий  $i$ -ую технологию взаимодействия компонентов ПС (таблица 3.7 Приложения 3);  
 $i = 1 \div m$  – количество используемых технологий взаимодействия компонентов ПС.

3.1.8. На основе рассчитанной базовой трудоемкости разработки ПС ( $T_{баз}$ ) определяются трудоемкости отдельных подпроцессов разработки. На трудоемкость подпроцессов разработки оказывают влияние различные факторы. Влияние этих факторов на трудоемкость подпроцессов учитывается с помощью поправочных коэффициентов уровня подпроцесса.

3.1.9. Трудоемкость подпроцесса «Анализ требований к ПС» ( $T_1$ ) определяется по формуле:

$$T_1 = A \cdot T_{баз} \cdot K_{квал.ан} \cdot K_{опыт.ан} , \quad (3.8)$$

где  $T_{баз}$  – базовая трудоемкость разработки ПС, определяемая в соответствии с пунктом 0;  
 $K_{квал.ан}$  – поправочный коэффициент, учитывающий уровень квалификации аналитиков (таблица 4.1 Приложения 4);  
 $K_{опыт.ан}$  – поправочный коэффициент, учитывающий опыт работы аналитиков в данной предметной области (таблица 4.2 Приложения 4);

$A$  – поправочный коэффициент, рассчитанный на основе собранной статистики в подразделениях Банка России в ( $A = 0,01$ ).

3.1.10. Трудоемкость подпроцесса «Проектирование» ( $T_2$ ) определяется по формуле:

$$T_2 = A \cdot T_{\text{баз}} \cdot K_{\text{квал.пр}} \cdot K_{\text{опыт.пр}} \cdot K_{\text{ср.пр}}, \quad (3.9)$$

где  $T_{\text{баз}}$  – базовая трудоемкость разработки ПС, определяемая в соответствии с пунктом 0;

$K_{\text{квал. пр}}$  – поправочный коэффициент, учитывающий уровень квалификации проектировщиков (таблица 4.3 Приложения 4);

$K_{\text{опыт пр}}$  – поправочный коэффициент, учитывающий опыт работы проектировщиков с используемыми средствами проектирования (таблица 4.4 Приложения 4).

$K_{\text{ср пр}}$  – поправочный коэффициент, учитывающий используемые средства проектирования (таблица 4.5 Приложения 4);

$A$  – поправочный коэффициент, рассчитанный на основе собранной статистики в подразделениях Банка России в ( $A = 0,12$ ).

3.1.11. Трудоемкость подпроцесса «Программирование» ( $T_3$ ) определяется по формуле:

$$T_3 = A \cdot T_{\text{баз}} \cdot K_{\text{квал.прог}} \cdot K_{\text{ср}}, \quad (3.10)$$

где  $T_{\text{баз}}$  – базовая трудоемкость разработки ПС, определяемая в соответствии с 0;

$K_{\text{квал прог}}$  – поправочный коэффициент, учитывающий уровень квалификации программистов (таблица 4.6 Приложения 4);

$K_{\text{ср}}$  – поправочный коэффициент, учитывающий используемые среды разработки (таблица 4.7 Приложения 4);

$A$  – поправочный коэффициент, рассчитанный на основе собранной статистики в подразделениях Банка России в ( $A = 0,79$ ).

3.1.12. Трудоемкость подпроцесса «Тестирование» ( $T_4$ ) определяется по формуле:

$$T_4 = A \cdot T_{\text{баз}} \cdot K_{\text{квал.тест}} \cdot K_{\text{БД}} \cdot K_{\text{ср.тс}} \quad (3.11)$$

где  $T_{\text{баз}}$  – базовая трудоемкость разработки ПС, определяемая в соответствии с пунктом 0;

$K_{\text{квал тест}}$  – поправочный коэффициент, учитывающий уровень квалификации персонала, осуществляющего тестирование (таблица 4.8 Приложения 4);

$K_{\text{ср. тс}}$  – поправочный коэффициент, учитывающий используемые средства тестирования (таблица 4.9 Приложения 4);

$K_{\text{БД}}$  – коэффициент, учитывающий размер БД (таблица 4.10 Приложения 4);

$A$  – поправочный коэффициент, рассчитанный на основе собранной статистики в подразделениях Банка России ( $A = 0,07$ ).

3.1.13. Трудоемкость подпроцесса «Ввод в действие ПС» ( $T_5$ ) определяется по формуле:

$$T_5 = A \cdot T_{\text{баз}} \cdot K_{\text{квал.вн}}, \quad (3.12)$$

где  $T_{\text{баз}}$  – базовая трудоемкость разработки ПС, определяемая в соответствии с пунктом 0;

$K_{\text{квал. вн}}$  – поправочный коэффициент, учитывающий уровень квалификации персонала, осуществляющего ввод в действие ПС (таблица 4.11 Приложения 4);

$A$  – поправочный коэффициент, рассчитанный на основе собранной статистики в подразделениях Банка России ( $A = 0,01$ ).

3.1.14. Итоговая трудоемкость процесса разработки ПС ( $T_{\text{разр}}$ ) складывается из трудоемкостей отдельных подпроцессов разработки ( $T_1, T_2, T_3, T_4, T_5$ )

$$T_{\text{разр}} = \sum_{i=1}^u T_i$$

(3.13)

где  $T_i$  – трудоемкость  $i$ -го подпроцесса разработки ПС.

$i = 1 \div u$  – количество подпроцессов разработки ПС (в Методике  $u = 5$ ).

3.1.15. Увеличение сроков разработки по сравнению с расчетными, как правило, не приводит к изменению суммарных трудозатрат на разработку ПС. Однако, в случае сокращения сроков выполнения работ по разработке ПС, существует тенденция к увеличению трудозатрат. В этом случае трудоемкость процесса разработки ПС следует скорректировать по следующей формуле:

$$T_{\text{срок}} = T \cdot K_{\text{ср}}$$

(3.14)

где  $T_{\text{срок}}$  – трудоемкость с учетом сроков разработки;

$T$  – трудоемкость процесса разработки;

$K_{\text{ср}}$  – коэффициент, учитывающий влияние сроков работ на трудоемкость (таблица 3.8 Приложения 3).

#### 4. МЕТОДИКА ОЦЕНКИ ЗАТРАТ ВРЕМЕНИ И ЧИСЛЕННОСТИ СПЕЦИАЛИСТОВ

4.1. Исходя из рассчитанных трудоемкостей подпроцессов разработки ПС, в зависимости от задаваемых ограничений, оцениваются сроки работ или количество необходимого персонала.

Ограничением может быть:

– количество персонала (используется при оценке сроков разработки ПС с учетом штата сотрудников);

– заданные сроки работ (используется при оценке необходимого числа сотрудников для реализации подпроцессов разработки или работ по сопровождению или эксплуатации ПС).

Срок выполнения подпроцесса разработки ПС ( $t$ , в месяцах) определяется по формуле:

$$t = \frac{T}{N \cdot \Phi}, \quad (4.1)$$

где  $T$  – трудоемкость подпроцесса разработки ПС [чел.-дни], для которого выполняется оценка сроков реализации;

$N$  – количество разработчиков, принимающих участие в данном подпроцессе разработки ПС;

$\Phi$  – фонд рабочего времени персонала в течение месяца, [дней/месяц];  
Срок реализации процесса разработки ПС ( $t_{разр}$ , в месяцах) определяется по формуле:

$$t_{разр} = \sum_{i=1}^u t_i \quad (4.2)$$

где  $t_i$  – время, реализации  $i$ -го подпроцесса разработки ПС, [месяцы];  
 $i = 1 \div u$  – количество подпроцессов разработки ПС (в Методике  $u = 5$ ).

Численность персонала для каждого подпроцесса разработки ПС ( $N$ , чел.) определяется по формуле:

$$N = \frac{T}{t \cdot \Phi} \quad , \quad (4.3)$$

где  $t$  – время, реализации подпроцесса разработки ПС [месяцы], для которого(ой) выполняется оценка численности персонала;

$T$  – трудоемкость данного подпроцесса разработки ПС [чел.-дни];

$\Phi$  – фонд рабочего времени персонала в течение месяца, [дней/месяц];

## ПРИЛОЖЕНИЕ 1

### КАТАЛОГ ФУНКЦИЙ ПС И ТАБЛИЦЫ ПЕРЕВОДНЫХ КОЭФФИЦИЕНТОВ

Каталог функций программных средств

Таблица 1.1

№	Наименование типа, класса; наименование (содержание) функции	Объем функции ПС (строк исходного текста на условном языке)
<b>1.</b>	<b>Управляющие операции</b>	
<b>1.1.</b>	<b><i>Информационная безопасность</i></b>	
1.1.1.	Реализация криптографических алгоритмов ( <i>один алгоритм</i> )	100-1000*
1.1.2.	Обеспечение безопасности передачи сообщений и обмена данными (шифрование)	200-1500
1.1.3.	Организация криптозащиты с помощью ПТК Криптоцентр авизо	5000-6000
1.1.4.	Контроль целостности ПО	2500-3500
1.1.5.	Реализация базовых функций информационной безопасности в соответствии с нормативными документами Банка России	6000-15000**
1.1.6.	Поддержка дискреционного контроля доступа	2000-5000
1.1.7.	Поддержка мандатного контроля доступа	2000-5000
1.1.8.	Создание, редактирование, удаление пользователей (группы пользователей)	1500-5000
1.1.9.	Установка прав доступа к объекту	1000-2000
1.1.10.	Ведение протокола доступа	2000-4000
1.1.11.	Настройка политики безопасности	1000-2000
1.1.12.	Ведение журналов отказа	1000-1500
1.1.13.	Взаимодействие с ПО Вербма-OW, САЭД	200-500
1.1.14.	Взаимодействие с ПО СКАД «Сигнатура»	200-1000
<b>1.2.</b>	<b><i>Реализация интерфейсов между программными средствами</i></b>	
1.2.1.	Взаимодействие с другим ПО	500-1000
1.2.2.	Взаимодействие с транспортной (почтовой) системой	2000-3000
1.2.3.	Реализация ПО обмена информацией между ЦОИ и клиентами (WEB-сервер)	9000-10000**
1.2.4.	Организация доступа к динамическим информационным ресурсам через Intranet Банка России (в т.ч. через КПИ Банка России)	10000-15000**
1.2.5.	Организация регулярно актуализируемого статистического контента для доступа через Intranet Банка России (в т.ч. через КПИ Банка России)	10000-15000**
<b>1.3.</b>	<b><i>Выполнение регламентных операций</i></b>	
1.3.1.	Пусковое решение	500-1000
1.3.2.	Регламентные операции: открытие и закрытие операционного дня в регионе, УБР, ответисполнителя (день, месяц, год)	4000-5000

1.3.3.	Операционный день	10000-15000
1.3.4.	Реализация процедур обновления версий ПО	1000-2000
1.3.5.	Управление режимами взаимодействия клиентов с КЦОИ	
1.3.6.	Управление режимами работы счета клиента в КЦОИ	
1.3.7.	Проведение автоматизированных процедур обработки учетно-операционной и расчетной информации ТУ в соответствии с регламентом функционирования ТУ	
1.3.8.	Управление автоматическими процедурами обработки, проводимыми по установленному регламенту, либо определяемых параметрами функционирования ТУ	
1.3.9.	Проведение сеанса урегулирования	
1.3.10.	Проведение регулярного сеанса	
1.3.11.	Проведение сеанса завершения обработки	
<b>1.4.</b>	<b><i>Реализация взаимосвязи программных средств и компонентов</i></b>	
1.4.1.	Сетевая передача команд и сообщений	100-1000
1.4.2.	Реализация связи между распределенными приложениями с использованием стандартных транспортных средств	100-1000
1.4.3.	Реализация связи между распределенными приложениями на основе сетевых интерфейсов низкого уровня	200-500
1.4.4.	Удаленная доставка информации с подтверждением получения	200-500
1.4.5.	Вызов удаленных процедур ( <i>единичный вызов</i> )	500-1000
1.4.6.	Управление файлами, доступом к файлам и передачей файлов между удаленными и разнородными файловыми системами	200-500
1.4.7.	Обработка сообщений	10-200
1.4.8.	Обработка распределенных транзакций	10-300
1.4.9.	Реализация средства контроля состояния распределенной сети однородных компонент	100-200
1.4.10.	Доступ к общей памяти: - в рамках одной машины - в рамках вычислительной сети	10-50 10-200
1.4.11.	Ведение журнала обращений к распределенному ПС	10-500
<b>1.5.</b>	<b><i>Создание и поддержка аналитических БД***</i></b>	
1.5.1.	Формирование физической структуры аналитических БД ( <i>на одну БД</i> )	1000-8000
1.5.2.	Формирование многомерной структуры аналитических БД ( <i>на одну БД</i> )	1000-8000
1.5.3.	Описание связей между аналитическими базами данных	1500-2000
1.5.4.	Формирование запросов к аналитическим БД ( <i>один запрос</i> )	10-80*
1.5.5.	Ведение журнала операций с аналитическими БД	1000-2000
1.5.6.	Формирование ETL-процедур	2000-15000
1.5.7.	Протоколирование ETL-процедур	2500-5000
1.5.8.	Функции настройки аналитических БД	1500-2000
1.5.9.	Модификация аналитических БД ( <i>один запрос</i> )	50-400*
1.5.10.	Контроль целостности и восстановление аналитических БД	500-1000
1.5.11.	Прикладное администрирование аналитических БД	1000-5000**

1.5.12.	Поиск, предоставление и вывод информации	1000-6000
1.5.13.	Обработка записей базы данных	500-3000
<b>2.</b>	<b>Вычислительные операции</b>	
<b>2.1.</b>	<b>Ввод и обработка данных</b>	
2.1.1.	Ввод данных первичных документов в интерактивном режиме	2000-9000
2.1.2.	Ввод данных по нарушениям в интерактивном режиме	2000-4000
2.1.3.	Ввод и первичный контроль документов из файлов	1500-6000
2.1.4.	Логический, синтаксический и номенклатурный контроль данных	1500-6000
2.1.5.	Расчет алгебраических выражений	100-1000
2.1.6.	Статистическая обработка данных	2000-5000
2.1.7.	Обработка документов на исключение из картотеки в ЦОИ	2000-2500
2.1.8.	Обработка документов ответисполнителей	4000-6000
2.1.9.	Обработка документов ответного авизования	6000-7000
2.1.10.	Обработка документов по принудительной квитанлке	1500-2000
2.1.11.	Обработка и учет заявок сопровождаемых регионов	2500-3500
2.1.12.	Обработка протоколов	2000-2500
2.1.13.	Обработка документов начального авизования	7000-10000
2.1.14.	Обработка документов МЭР в ЦОИ	5000-10000
2.1.15.	Ввод прочих данных	1000-4000
2.1.16.	Обработка входной информации (прочих данных)	1500-6000
2.1.17.	Оплата платежей по МЭР/по расчетам с ПУ ТУ	
2.1.18.	Обработка ошибочных платежей	
2.1.19.	Реализация в ПС УБР функций режима резервного ввода	
2.1.20.	Ввод реквизитов расчетных документов на бумажных носителях	
2.1.21.	Ввод документов для учета операций по внебалансовым счетам	
2.1.22.	Ввод документов для учета операций по внутрибанковским и кассовым операциям	
<b>2.2.</b>	<b>Составление и анализ балансов</b>	
2.2.1.	Составление балансов	500-1000**
2.2.2.	Анализ балансов	1000-3500**
2.2.3.	Выполнение баланса и оборотов КО за месяц	2500-3500**
2.2.4.	Выполнение ежедневного баланса	1500-2000**
2.2.5.	Корректировка балансов	200-500**
2.2.6.	Пересчет балансов	2000-3000**
<b>2.3.</b>	<b>Проведение экономического анализа</b>	
2.3.1.	Анализ безналичных расчетов в платежных системах	2500-3500**
2.3.2.	Анализ денежной массы в обращении по районам	5000-6000**
2.3.3.	Анализ курсов валют, устанавливаемых ЦБ	3500-4500**
2.3.4.	Анализ курсов основной валюты в обменных пунктах КО	7000-8000**
2.3.5.	Анализ событий нефинансового характера	300-1100**
<b>2.4.</b>	<b>Выполнение задач планирования</b>	
2.4.1.	Планирование инспекторских проверок	1500-2000**
2.4.2.	Планирование эмиссии	4000-5000**
2.4.3.	Планирование прочих задач	800-4000**

<b>2.5.</b>	<b><i>Мониторинг</i></b>	
2.5.1.	Мониторинг отчетности КО	6000-7000**
2.5.2.	Мониторинг отчетности ТУ	6000-7000**
<b>2.6.</b>	<b><i>Формирование и обработка аналитических показателей</i></b>	
2.6.1.	Расчет экономических показателей	5000 – 15000**
2.6.2.	Экономический и финансовый анализ	10000 – 20000**
2.6.3.	Моделирование и прогнозирование	20000 – 30000**
2.6.4.	Ведение показателей системы, определяющих ее штатное состояние функционирования	
<b>3.</b>	<b><i>Операции, зависящие от аппаратуры</i></b>	
<b>3.1.</b>	<b><i>Прием документов от различных источников</i></b>	
3.1.1.	Прием файлов, первичный контроль	300-1500
3.1.2.	Прием информации из ЦОИ	2000-8000
3.1.3.	Прием данных, содержащихся в DBF файлах	200-1200
3.1.4.	Прием описи к конвертам с документами	1500-2000
3.1.5.	Прием текстовых файлов	300-800
<b>3.2.</b>	<b><i>Регистрация входных документов</i></b>	
3.2.1.	Регистрация входных документов	500-1000
3.2.2.	Регистрация исходящих авизо	2500-3000
3.2.3.	Регистрация телеграфных и почтовых авизо	1000-1500
<b>3.3.</b>	<b><i>Работа с файлами и форматами</i></b>	
3.3.1.	Загрузка файлов	1000-1500
3.3.2.	Преобразование формата данных	1000-4500
3.3.3.	Формирование файлов	500-3000
<b>3.4.</b>	<b><i>Ведение архива и копирование информации</i></b>	
3.4.1.	Создание архива	500-2500
3.4.2.	Добавление данных в архив	500-1000
3.4.3.	Извлечение данных из архива	500-1000
3.4.4.	Поиск данных в архиве	500-1500
3.4.5.	Проверка целостности архива	500-1500
3.4.6.	Задачи ведения электронного архива	2500-7500**
<b>4.</b>	<b><i>Операции управления данными</i></b>	
<b>4.1.</b>	<b><i>Ведение журналов</i></b>	
4.1.1.	Ведение журналов ( <i>на один журнал</i> )	500-5000*
4.1.2.	Ведение журнала регистрации запросов и подтверждений по начальным (ответным) авизо	2000-5000
4.1.3.	Вывод журналов приема-передачи	500-800
4.1.4.	Контроль и журнализация доступа к защищенным ресурсам	1500-2000
<b>4.2.</b>	<b><i>Работа со справочниками</i></b>	
4.2.1.	Ведение справочников ( <i>на один справочник</i> )	500-3000*
4.2.2.	Ведение книги регистрации открытых лицевых счетов	10000 - 10800
<b>4.3.</b>	<b><i>Контроль информации документов</i></b>	
4.3.1.	Контроль входной информации документов	500-2000
4.3.2.	Контроль информации при вводе данных документа ( <i>на форму ввода</i> )	300-400*
4.3.3.	Контроль МЭР/расчетов с ПУ ТУ	

<b>4.4.</b>	<b><i>Настройка ПС</i></b>	
4.4.1.	Разработка и ввод метаданных	1000-8000
<b>4.5.</b>	<b><i>Ведение и обработка протоколов</i></b>	
4.5.1.	Протоколирование работы	2500-3500
4.5.2.	Ведение протокола выполнения расчетов	500-700
4.5.3.	Формирование протокола проводок	700-1500
4.5.4.	Обработка протоколов	1000-2000
<b>4.6.</b>	<b><i>Формирование отчетов</i></b>	
4.6.1.	Формирование отчетов ( <i>на отчет</i> )	100-4000*
4.6.2.	Просмотр и редактирование отчетных форм	500-1000
4.6.3.	Подготовка форм отчетности ( <i>на одну форму</i> )	1000-3000*
<b>4.7.</b>	<b><i>Распределенная обработка данных</i></b>	
4.7.1.	Создание одного объекта на базе технологии CORBA	50*
4.7.2.	Создание одного объекта на базе технологии COM	60*
4.7.3.	Вызов метода CORBA-объекта ( <i>один вызов</i> )	2*
4.7.4.	Вызов метода COM-объекта ( <i>один вызов</i> )	1*
4.7.5.	Реализация сетевого взаимодействия на базе средств Sockets API (серверная сторона, один сервер)	40*
4.7.6.	Реализация сетевого взаимодействия на базе средств Sockets API (клиентская сторона, один клиент)	20*
<b>4.8.</b>	<b><i>Ведение очереди отложенных платежей</i></b>	
4.8.1.	Централизованное ведение очереди отложенных платежей	
4.8.2.	Управление очередью отложенных платежей	
<b>4.9.</b>	<b><i>Ведение картотек расчетных документов</i></b>	
4.9.1.	Ведение картотеки расчетных документов, ожидающих акцепта для оплаты	
4.9.2.	Ведение картотеки расчетных документов, не оплаченных в срок	
4.9.3.	Прием ЭСИС – запросов УБР для получения информации об актуальном состоянии картотек	
4.9.4.	Ведение картотек расчетных документов	
<b>4.10.</b>	<b><i>Ведение синтетического и аналитического учета</i></b>	
4.10.1.	Отражение операций по лицевым счетам аналитического учета	28000
4.10.2.	Ведение синтетического учета совершаемых операций	16875
<b>5.</b>	<b><i>Операции управления пользовательского интерфейса</i></b>	
<b>5.1.</b>	<b><i>Настройка ПС на условия применения</i></b>	
5.1.1.	Настройка параметров	500-1000
5.1.2.	Настройка аналитических таблиц	1000-2000
<b>5.2.</b>	<b><i>Реализация пользовательского интерфейса<sup>1</sup></i></b>	
5.2.1.	Реализация интерфейса пользователя	20-5000
5.2.2.	Реализация стандартного графического пользовательского интерфейса (однооконное приложение)	300-2000
5.2.3.	Реализация стандартного графического пользовательского интерфейса (диалоговое приложение)	20-1000
5.2.4.	Реализация стандартного графического пользовательского	500-7000

	интерфейса (многооконное приложение)	
5.2.5. 5.2.6.	Реализация стандартного графического пользовательского интерфейса (однооконное приложение) с использованием: – API – MFC/OWL	1000–2000 600–900
5.2.8. 5.2.9. 5.2.10.	Реализация стандартного графического пользовательского интерфейса (диалоговое приложение) с использованием: – API – MFC/OWL – VCL	500–1000 40-100 20-100
5.2.11. 5.2.12. 5.2.13.	Реализация стандартного графического пользовательского интерфейса (многооконное приложение) с использованием: – API – MFC/OWL – VCL	2000–7000 600–1500 500–1500
5.2.14.	Реализация стандартного графического пользовательского интерфейса (однооконное приложение) с использованием WEB-технологий	300-12000
<b>5.3.</b>	<b><i>Реализация машинной графики</i></b>	
5.3.1.	Реализация машинной графики для отображения состояния ПС в статике	500-1500
5.3.2.	Реализация машинной графики для отображения состояния ПС в динамике	500-3000
5.2.7.	– VCL	300-700

\* Функции выполняют единичную операцию и могут иметь несколько реализаций как на уровне ПС в целом, так и на уровне компонентов ПС. Каждая реализация должна характеризоваться коэффициентом степени повторного использования.

\*\* Функции являются композитными (либо реализующими классы задач) и включают в себя другие функции из Каталога. Их использование допускается только для крупных ПС, а также для прочих ПС, в которых трудно детализировать набор функций. Такие функции не могут реализовываться более одного раза в расчете. Детализация композитных функций приведена в Рекомендациях к Методике.

Все остальные функции являются структурными единицами ПС. Могут иметь несколько реализаций на уровне компонентов ПС и одну реализацию на уровне ПС. Каждая реализация должна характеризоваться коэффициентом степени повторного использования.

\*\*\* Допускается применение функций данной группы к базам данных, не являющихся аналитическими.

<sup>1</sup> Под одним пользовательским интерфейсом следует понимать всю совокупность окон, элементов управления и прочих элементов, реализующих интерфейс одного ПС. Таким образом, одно ПС может реализовывать только одну функцию из группы не более одного раза.

Значения переводных коэффициентов ( $kp_i$ ) для различных средств разработки

Таблица 1.2

Средство разработки*	Значение $kp_i$
Языки низкого уровня (Assembler)	0,17
Pascal	0,42
C#	1,64
C++	1
VBA	3,48
Clarion	2,56
Встроенные в ПО макроязыки (1С)	3,86
Командные скриптовые языки	1,87
HTML/CSS	0,15
XML/XSLT	0,2
Java	1,69
JavaScript	1,87
ASP	1,74
SQL	1,96
C	0,51
Object Pascal	0,67
Visual Basic	3,48
Clipper	2,36
FoxPro	1,51
PL/1	0,91
PL/SQL	1,71
PHP	1,74
J2EE	1,06
JCL	1,1
JSP	1,02
Perl	1,74
Rexx	0,9
VBScript	1,56
Uniface	3.48

\*Примечание. Указанные в таблице средства разработки обозначают группу средств подобного типа. Поэтому если средство разработки непосредственно не указано в таблице, то его необходимо самостоятельно отнести к какой-либо из групп, наиболее близкой по уровню используемого языка.

Таблица переводных коэффициентов для расчета объема сопровождаемого ПС  
через Каталог функций

Таблица 1.3

Средство разработки*	Коэффициент А	Коэффициент С
C++	1	1
VBA	1.024	1.0076
Clarion	1.07	0.9946
Встроенные в ПО макроязыки (1С)	1.024	1.0076
HTML/CSS	1	1
Java	1.105	0.9911
Object Pascal	1.024	1.0076
Visual Basic	1.024	1.0076
Clipper	1.07	0.9946
FoxPro	8.001	0.8243
PL/SQL	1.1757	0.9886
Uniface	1.024	1.0076

*\*Примечание:* Указанные в таблице средства разработки обозначают группу средств подобного типа. Поэтому если средство разработки непосредственно не указано в таблице, то его необходимо самостоятельно отнести к какой-либо из групп, наиболее близкой по уровню используемого языка. Если сделать это невозможно (либо затруднительно), то поправочные коэффициенты А и С принять равными единице.

**ПОПРАВочНЫЕ КОЭФФИЦИЕНТЫ УРОВНЯ ФУНКЦИИ**

Уровень сложности управляющих операций

Таблица 2.1

Уровень сложности	Характеристика уровня	Значение $L_v$
Очень низкий	Код без циклов, с небольшим количеством не вложенных структурированных операторов. Упрощенная схема взаимодействия модулей посредством вызова процедур	1
Низкий	Простое вложение структурированных операторов. Применение, в основном, простых предикатов	2
Средний	Применение, в основном, простого вложения и нескольких средств межмодульного взаимодействия, логических таблиц, простых функций обратного вызова или обмена сообщениями, включая распределенную обработку на основе промежуточного ПО	3
Высокий	Множественное вложение структурированных операторов с множеством составных предикатов. Управление очередью и стеком. Однородная распределенная обработка. Мягкие требования к режиму однопроцессорной обработки в реальном времени	4
Очень высокий	Использование методов написания программ на основе рекурсии и многократного входа в программу. Обработка прерываний фиксированного приоритета, синхронизация задач, сложные функции обратного вызова, разнородная распределенная обработка. Жесткие требования к режиму однопроцессорной обработки в реальном времени	5
Сверхвысокий	Сложное управление ресурсами с динамически изменяющимися приоритетами. Управление на уровне микрокоманд. Жесткие требования к режиму распределенной обработки	6

*Примечание.* В том случае, когда характеристика уровня соответствует нескольким положениям из таблиц 2.1-2.5, то выбирается наиболее высокий уровень сложности

### Уровень сложности вычислительных операций

Уровень сложности	Характеристика уровня	Значение $L_v$
Очень низкий	Вычисление упрощенных выражений: например, $A=B+C*(D-E)$	1
Низкий	Вычисление выражений средней сложности (одномерные массивы)	2
Средний	Использование стандартных математических и статистических процедур. Основные операции с матрицами/векторами	3
Высокий	Базовые элементы численного анализа: многомерная интерполяция, обыкновенные дифференциальные уравнения, простые случаи усечения и округления	4
Очень высокий	Сложный, но структурированный численный анализ, матричные уравнения близкие к сингулярным, дифференциальные уравнения в частных производных. Простое распараллеливание	5
Сверхвысокий	Сложный и неструктурированный численный анализ: высокоточный анализ стохастических данных. Сложное распараллеливание	6

Таблица 2.2

### Уровень сложности операций, зависящих от аппаратуры

Таблица 2.3

Уровень сложности	Характеристика уровня	Значение $L_v$
Очень низкий	Упрощенные операторы чтения, записи с простыми форматами	1
Низкий	Не требуется никакой информации о характеристиках конкретного типа процессоров или устройств ввода/вывода. Ввод/вывод осуществляется на уровне команд типа GET/PUT	2
Средний	Операции ввода/вывода включают выбор устройства, проверка его состояния и обработку ошибок	3
Высокий	Операции ввода/вывода на физическом уровне (трансляция физических адресов хранения данных; операции поиска, чтения, и т.д.). Оптимизированное совмещение ввода/вывода	4
Очень высокий	Процедуры для определения, обработки и маскирования прерываний. Управление каналом связи. Встроенные системы с определенными требованиями к производительности	5
Сверхвысокий	Написание управляющего кода, работающего синхронно с устройством, программирование на уровне микроопераций. Встроенные системы, критичные к производительности	6

Уровень сложности операций управления данными

Таблица 2.4

Уровень сложности	Характеристика уровня	Значение $L_v$
Очень низкий	Простые массивы в основной памяти. Простые запросы на обновление	1
Низкий	Использование единственного файла без изменения структуры данных, без редактирования. Промежуточные файлы не используются. Запросы на обновление и запросы SQL умеренной сложности	2
Средний	Многофайловый ввод и однофайловый вывод. Простые структурные изменения, простые правки. Сложные запросы на обновление и запросы SQL	3
Высокий	Простые триггеры, активизируемые содержанием потоков данных. Сложное реструктурирование данных	4
Очень высокий	Управление распределенными базами данных. Сложные триггеры. Оптимизация поиска	5
Сверхвысокий	Сильно связанные динамические реляционные и объектные структуры. Речевое управление данными	6

Уровень сложности операций управления пользовательского интерфейса

Таблица 2.5

Уровень сложности	Характеристика уровня	Значение $L_v$
Очень низкий	Простые формы, генераторы отчетов	1
Низкий	Использование простых средств построения интерфейса пользователя	2
Средний	Использование простых (стандартных) элементов управления	3
Высокий	Разработка новых элементов управления и усовершенствование существующих. Простой голосовой ввод/вывод, мультимедиа	4
Очень высокий	Двух/трехмерная графика средней сложности, динамическая графика, мультимедиа	5
Сверхвысокий	Сложная мультимедийная информация, виртуальная реальность, интерфейс речевого управления	6

Таблица 2.6

Значение коэффициента сложности функции ( $K_{сложн}$ ) в зависимости от уровня сложности функции

Уровень сложности функции	$L_v$	Значение $K_{сложн}$
Очень низкий	1	0,82
Низкий	2	0,87
Средний	3	1,00
Высокий	4	1,10
Очень высокий	5	1,20
Сверхвысокий	6	1,41

Значения коэффициента, учитывающего конкретные условия и языки программирования ПС  
( $K_{ср.разр}$ )

Таблица 2.7

Средство разработки*	Значения $K_{ср.разр}$
Языки низкого уровня (Assembler)	1,52
Pascal	1,18
C#	0,92
C++	1
VBA	0,69
Clarion	0,47
Встроенные в ПО макроязыки (1С)	0,73
Командные скриптовые языки	0,79
HTML/CSS	0,64
XML/XSLT	0,69
Java	0,92
JavaScript	0,76
ASP	0,81
SQL	0,65
C	1,09
Object Pascal	1,1
Visual Basic	0,69
Clipper	0,58
FoxPro	0,66
PL/1	1,09
PL/SQL	0,58
PHP	0,81
J2EE	0,94
JCL	0,91
JSP	0,98
Perl	0,81
Rexx	1,12
VBScript	0,4
Uniface	0.69

*Примечание.* Указанные в таблице средства разработки обозначают группу средств подобного типа. Поэтому если средство разработки непосредственно не указано в таблице, то его необходимо самостоятельно отнести к какой-либо из групп, наиболее близкой по уровню используемого языка.

Значения коэффициента, учитывающего опыт работы программистов с данным средством разработки ( $K_{опыт}$ )

Таблица 2.8

Уровень опыта работы разработчика	Опыт работы разработчика	Значение $K_{опыт}$
Низкий	1-2 разработанных ПС (компонентов АС)*	1,09
Средний	3-4 разработанных ПС (компонентов АС)*	1,00
Высокий	5 разработанных ПС (компонентов АС)*	0,91
Очень высокий	более 5 разработанных ПС (компонентов АС)*	0,84

*Примечание.* Учитывается количество разработанных ПС (компонентов АС) с учетом их развития.

**ПРИЛОЖЕНИЕ 3**

**ПОПРАВочНЫЕ КОЭФФИЦИЕНТЫ УРОВНЯ РАСЧЕТА ПО РАЗРАБОТКЕ  
ПРОГРАММНОГО СРЕДСТВА**

Значение коэффициента, учитывающего степень новизны ПС ( $K_n$ )

Таблица 3.1

Степень новизны	Значение $K_n$
Принципиально новое ПС, разрабатываемое для нового типа ТС/ОС*	1,42
Принципиально новое ПС, разрабатываемое для прежнего типа ТС и нового типа ОС	1,34
Принципиально новое ПС, разрабатываемое для нового типа ТС и прежнего типа ОС	1,12
Принципиально новое ПС, разрабатываемое для прежнего типа ТС/ОС	1,06
ПС, являющееся развитием определенного параметрического ряда ПС на новом типе ТС/ОС	1,00
ПС, являющееся развитием определенного параметрического ряда ПС на прежнем типе ТС и новом типе ОС	0,94
ПС, являющееся развитием определенного параметрического ряда ПС на новом типе ТС и прежнем типе ОС	0,87
ПС, являющееся развитием определенного параметрического ряда ПС на прежнем типе ТС/ОС	0,80

\*Примечание. Термин «Новый тип ТС/ОС» означает, что данный тип ТС/ОС не использовался в подразделении Банка России.

Значения коэффициента, учитывающий требования к надежности ПС ( $K_{над}$ )

Таблица 3.2

Уровень требований к надежности ПС	Характеристика	Значение $K_{над}$
Низкий	Незначительный, легко восполнимый ущерб	0,95
Средний	Средний, восполнимый ущерб	1,00
Высокий	Сбои ПС приводят к потерям рабочей информации	1,06
Очень высокий	Потери важной информации	1,16

Значения коэффициента, учитывающего требования к производительности ПС ( $K_{произв}$ )

Таблица 3.3

Характеристика	Значение $K_{произв}$
Производительность ПС не играет роли	0,94
Требования к производительности ПС не установлены (однако производительность ПС должна обеспечивать приемлемое время отклика при работе пользователя в интерактивном режиме)	1,00
Имеются умеренные требования к производительности	1,06
Повышенные требования к производительности	1,12

Исключительно высокие требования к производительности	1,18
---	------

Значения коэффициента, учитывающего требования к уровню информативности документации ( $K_{докум}$ )

Таблица 3.4

Уровень требований	Характеристика	Значение $K_{докум}$
Низкий	Не учтены некоторые потребности жизненного цикла	0,94
Средний	Соответствует потребностям жизненного цикла	1,00
Высокий	Повышенный объем для жизненного цикла данного ПС	1,07
Очень высокий	Большой (избыточный) объем для жизненного цикла данного ПС	1,14

*Примечание.* Экономия трудозатрат путем установления очень низкого значения коэффициента повлечет дополнительные расходы в процессе сопровождения.

Значения коэффициента, учитывающий опыт разработки программных средств подобного типа ( $K_{оп}$ )

Таблица 3.5

Уровень опыта разработки приложений	Опыт разработки	Значение $K_{оп}$
Низкий	1-2 разработанных ПС (компонентов АС)*	1,10
Средний	3-4 разработанных ПС (компонентов АС)*	1,00
Высокий	5 разработанных ПС (компонентов АС)*	0,88
Очень высокий	более 5 разработанных ПС (компонентов АС)*	0,81

*Примечание.* Учитывается количество разработанных ПС (компонентов АС) с учетом их развития.

Значения коэффициента, учитывающего структуру разрабатываемого программного средства ( $K_{стр}$ )

Таблица 3.6

Тип структуры	Значения $K_{стр}$
Монолитное ПС	0,70
ПС с набором библиотек	1,00
Многокомпонентное ПС, функционирующее в рамках одной рабочей станции	1,10
Многокомпонентное ПС, работающее в соответствии с архитектурой «клиент-сервер»	1,25
Распределенное программное средство	1,60
Распределенное программное средство, функционирующее на неоднородном аппаратном обеспечении	1,85

Значения коэффициента, учитывающего технологию взаимодействия компонентов ПС ( $K_m$ )

Таблица 3.7

Технологии	Значения $K_m$
Оригинальный протокол взаимодействия (sockets)	0,80
Использование Интернет протоколов	0,40

Удаленный вызов процедур (RPC)	0,45
Брокер запросов (CORBA)	0,30
Объектно-ориентированные технологии (COM/DCOM, ODBC, OLE DB, ADO, ActiveX)	0,35

Значения коэффициента, учитывающего влияние сроков работ на трудоемкость ( $K_{cp}$ )

Таблица 3.8

Продолжительность работ по сравнению с номинальной, %	Значение $K_{cp}$
Меньше или равно 75	1,43
от 76 и меньше 85	1,14
от 86 и меньше 100	1,05
Равно или более 100	1,00

*Примечание.* Под номинальной следует понимать продолжительность работ с учетом имеющегося персонала и его опыта при условии работы 8 часов в сутки, 5 дней в неделю.

**ПРИЛОЖЕНИЕ 4**

**ПОПРАВочНЫЕ КОЭФФИЦИЕНТЫ УРОВНЯ ПОДПРОЦЕССА РАЗРАБОТКИ  
ПРОГРАММНОГО СРЕДСТВА**

Значения коэффициента, учитывающего уровень квалификации аналитиков ( $K_{\text{квал ан}}$ )

Таблица 4.1

Уровень квалификации аналитиков	Значение $K_{\text{квал ан}}$
Средний	1,00
Высокий	0,87
Очень высокий	0,74
Исключительно высокий	0,66

*Примечание.* Аналитики – часть персонала организации, занятая в разработке технических требований, высокоуровневом проектировании и составлении рабочего плана.

Значения коэффициента, учитывающего опыт работы аналитиков в данной предметной области ( $K_{\text{опыт ан}}$ )

Таблица 4.2

Уровень опыта работы аналитика	Опыт работы аналитика	Значение $K_{\text{опыт ан}}$
Низкий	1-2 разработанных ПС (компонентов АС)*	1,07
Средний	3-4 разработанных ПС (компонентов АС)*	1,00
Высокий	5 разработанных ПС (компонентов АС)*	0,87
Очень высокий	более 5 разработанных ПС (компонентов АС)*	0,81

*Примечание.* Учитывается количество разработанных ПС (компонентов АС) с учетом их развития.

Значения коэффициента, учитывающего уровень квалификации проектировщиков ( $K_{\text{квал пр}}$ )

Таблица 4.3

Уровень квалификации проектировщиков	Значение $K_{\text{квал пр}}$
Средний	1,00
Высокий	0,92
Очень высокий	0,83
Исключительно высокий	0,77

Значения коэффициента, учитывающего опыт работы проектировщиков с используемыми средствами проектирования ( $K_{\text{опыт пр}}$ )

Таблица 4.4

Уровень опыта работы проектировщика	Опыт работы проектировщика	Значение $K_{\text{опыт пр}}$
Низкий	1-2 разработанных ПС (компонентов АС)*	1,04
Средний	3-4 разработанных ПС (компонентов АС)*	1,00
Высокий	5 разработанных ПС (компонентов АС)*	0,92
Очень высокий	более 5 разработанных ПС (компонентов АС)*	0,88

*Примечание.* Учитывается количество разработанных ПС (компонентов АС) с учетом их развития.

Значения коэффициента, учитывающего используемые средства проектирования ( $K_{ср пр}$ )

Таблица 4.5

Средства проектирования	Значения $K_{ср пр}$
Средства проектирования не использовались	1,06
Средства проектирования БД (BPWIN, ERWIN/ERX, Designer 2000, DataDirect Explorer, Oracle Designer, Silverrun RBM и BPM, Together J)	1,03
Средства проектирования типа Microsoft Visual Modeler, Cadre ОМТ	0,93
Объектно-ориентированные CASE- средства (Rational Rose, db-UIM/X)	0,84

Значения коэффициента, учитывающего уровень квалификации программистов ( $K_{квал прог}$ )

Таблица 4.6

Уровень квалификации программистов	Значение $K_{квал прог}$
Средний	1,00
Высокий	0,95
Очень высокий	0,89
Исключительно высокий	0,86

*Примечание.* Оценка должна строиться на возможностях программистов как членов группы, а не индивидуальных возможностях каждого программиста.

Значения коэффициента, учитывающего используемую среду разработки ( $K_{ср}$ )

Таблица 4.7

Используемая среда разработки	Значение $K_{ср}$
Простые средства редактирования (Notepad, vi), необходимость написания make-файлов	1,06
Мощные текстовые редакторы (UltraEdit, MultiEdit, XEmacs), средства автоматизации создания make-файлов (JAM)	1,04
Интегрированные среды разработки (Visual Studio)	1
Мощные интегрированные среды разработки с возможностью объектно-ориентированного проектирования и автоматической генерацией кода (db-UIM/X, Borland JBuilder, Borland C++ Builder, PowerBuilder, Borland Delphi)	0,95
Среды разработки на основе СУБД типа Oracle (Developer/FORMS), Informix, MS SQLServer, Paradox, dbase, FoxPro (Clarion)	0,92
Среды разработки с использованием VBA (или его аналогов) Microsoft Access, Microsoft Excel, 1C, Lotus Notes	0,90

Значения коэффициента, учитывающего уровень квалификации персонала, осуществляющего тестирование ( $K_{\text{квал тест}}$ )

Таблица 4.8

Уровень квалификации персонала, осуществляющего тестирование	Значение $K_{\text{квал тест}}$
Средний	1,00
Высокий	0,89
Очень высокий	0,76
Исключительно высокий	0,69

Значения поправочного коэффициента, учитывающего используемые средства тестирования ( $K_{\text{ср. тс}}$ )

Таблица 4.9

Средства тестирования	Значения $K_{\text{ср. тс}}$
Автоматизированные средства тестирования не использовались	1,00
Автоматизированные средства тестирования использовались	0,76

Значения коэффициента, учитывающего размер БД ( $K_{\text{БД}}$ )

Таблица 4.10

Размер БД	Отношение объема тестовых данных в БД к объему ПС, байты/количество строк исходного текста (D/P)	Значение $K_{\text{БД}}$
Низкий	$< 10$	0,95
Средний	$10 \leq D/P < 100$	1,0
Высокий	$100 \leq D/P < 1000$	1,08
Очень высокий	$D/P \geq 1000$	1,15

*Примечание.* Объем данных в БД рассчитывается исходя из размера одной записи и количества записей. Размер одной записи рассчитывается как сумма полей записи с учетом используемых типов полей.

Значения коэффициента, учитывающего уровень квалификации персонала, осуществляющего ввод в действие ПС ( $K_{\text{квал вн}}$ )

Таблица 4.11

Уровень квалификации персонала, осуществляющего ввод в действие ПС	Значение $K_{\text{квал вн}}$
Средний	1,00
Высокий	0,94
Очень высокий	0,88
Исключительно высокий	0,87

## ПРИЛОЖЕНИЕ 5

### РЕКОМЕНДАЦИИ К РАСЧЕТУ ТРУДОЕМКОСТИ ПРОГРАММНЫХ СРЕДСТВ

#### 5.1. ОЦЕНКА ОБЪЕМА ПС

Оценка объема разрабатываемого ПС выполняется с использованием Каталога функций программных средств (далее Каталог), приведенного в таблице 1.1 Приложения 1 Методики.

Состав функций разрабатываемого ПС формируется на основе Технического задания (ТЗ) на разработку ПС. Программному средству устанавливается в соответствие набор функций из Каталога. Данный набор функций должен соответствовать следующим основным требованиям:

- охватывать полностью все возможности ПС, описанные в ТЗ и пояснительной записке;
- не иметь пересечений: разные функции не должны выполнять одни и те же действия для решения одной задачи;
- состоять из функций с наибольшей детализацией.

**Правила применения Каталога представлены в пункте 3.1 Методики.**

Каталог содержит функции трех типов:

1 тип. Атомарные функции (обозначены в Каталоге символом \*). Например, функция 1.5.4 «Формирование запросов к аналитическим БД», объем которой задан исходя из выполнения одного запроса. Количество реализаций этой функции будет определяться числом различных выполняемых запросов.

2 тип. Композитные функции и функции, реализующие классы задач (обозначены в Каталоге символом \*\*). Например, функция 1.5.11 «Прикладное администрирование аналитических БД». Функции этого типа составлены из других функций Каталога в рамках того же класса функций или представляют собой реализацию значительной части программного средства. Например, приведенная функция включает в себя контроль целостности и восстановление БД, функции настройки и ведение журнала операций с БД (функции 1.5.5, 1.5.8, 1.5.10 Каталога). Объем таких функций примерно соответствует объему входящих в нее функций с учетом доли участия каждой из них в составе композитной функции (см. таблицу 2 Приложения 1 Рекомендаций).

Функция не может применяться одновременно с соответствующей композитной функцией, в состав которой она входит, для решения одной задачи. Например, Если для работы с базой данных выбрана функция «Прикладное администрирование аналитических БД», то недопустимо для работы с этой же БД использовать функцию «Контроль целостности и восстановление БД».

В этой связи применение композитных функций и функций, реализующих классы задач, допускается в следующих случаях:

- если для ПС затруднительно далее провести детализацию функций (например, недостаточно детально изложено техническое задание);
- для крупных ПС, требующих представления набора функций на более высоком уровне.

Допускается использование не более одной функции данного типа в рамках расчета. Для повышения точности рекомендуется использовать более детальные функции из Каталога. Детальными по отношению к композитным функциям являются функции первого и третьего типа.

3 тип. Функции, реализующие отдельную задачу. К ним относятся все остальные функции Каталога, не относящиеся к двум первым типам. Для таких функций допускается повторное использование.

Для определения объема разрабатываемого ПС необходимо определить объем каждой отдельной функции, входящей в набор функций данного ПС. Объем функции определяется одним из способов, приведенных в Методике в пункте 3.1.1.

Объемы функций в Каталоге представлены в виде диапазона значений. Нижняя граница объема функции из Каталога соответствует предоставлению минимально возможного набора операций для реализации задачи функции, что вполне достаточно для реализации ПС среднего уровня.

Если значение объема функции на языке разработки (например, в случае расчета по написанному коду ПС) находится ниже или выше рекомендуемого диапазона, то в качестве значения объема функции следует принимать нижнюю или верхнюю границу допустимого диапазона значений соответственно.

Выбор объема функции с верхней границей объема должен обязательно обосновываться разработчиком.

Если функция входит в состав готового программного модуля или библиотеки (например, разработчик использует готовый исходный код или реализацию функции из заимствованного программного обеспечения), то на трудозатраты она никак не влияет и степень ее повторного использования при этом равна 0 (полностью используется повторно). Например, степень повторного использования для любой функции из заимствованного программного модуля, включаемого в разрабатываемое программное средство, равна 0.

Если в рамках программного средства или компонента программного средства некоторая функция используется повторно (исходный код или реализация функции дописывается разработчиком, или исходный код готовой функции разработчиком частично модифицируется), то величина коэффициента повторного использования будет равна доле изменений объема функции, используемой повторно. Например, в расчете необходимо использовать чтение данных XML формата. При этом используется готовая библиотека, встроенная в операционную систему. Для полного достижения необходимой функциональности (если, например, после чтения XML необходимо определенным образом обработать данные) необходимо написать еще 500 строк кода. Объем функции «Загрузка файлов», с помощью которой моделируется чтение (загрузка) XML формата, составляет по каталогу 1250 строк кода. Таким образом, степень повторного использования функции составит  $k_i = 500 / 1250 = 0.4$ .

В случае, когда функция реализуется неоднократно (число реализаций функции задается больше 1), величина коэффициента ее повторного использования (для заданного числа реализаций) определяется как доля изменений объема всех реализаций. Например, для функции «Ведение журналов» с объемом 2750 строк кода задано 4 реализации ( $r_i = 4$ ) по одной на тип журнала. Объем всех реализаций функции без учета прочих факторов составит  $2750 * 4 = 11000$  строк кода. При этом 40% кода каждой функции ( $2750 * 0.4 = 1100$  строк кода) одинаковы (используется повторно). Объем повторно используемого кода составит  $1100 * 4 = 4400$  строк кода. Таким образом, степень повторного использования составит  $k_i = (11000 - 4400) / 11000 = 0,6$  (60%).

**Объем программного средства в строках исходного кода с учетом его структуры рассчитывается в соответствии с пунктом 3.1.5 Методики.**

## 5.2. ОЦЕНКА ТРУДОЕМКОСТИ РАЗРАБОТКИ ПРОГРАММНОГО СРЕДСТВА

Расчет трудоемкости разработки программных средств выполняется Разработчиком ПС.

Оценка трудоемкости разработки программного средства выполняется в два этапа:

- *расчет базовой трудоемкости разработки ПС;*
- *расчет трудоемкости отдельных подпроцессов разработки;*

Расчет базовой трудоемкости разработки выполняется в соответствии с пунктом 3.2.6 Методики.

Расчет трудоемкости отдельных подпроцессов разработки ПС выполняется строго в соответствии с пунктами 3.1.8–3.1.14 Методики. Поскольку модель расчета трудоемкости разработки предполагает разбиение базового значения трудоемкости по этапам, то итоговая величина трудоемкости разработки считается путем суммирования ее по всем этапам (пункт 3.1.14 Методики).

Расчет трудоемкости разработки с учетом влияния сроков работ выполняется в соответствии с пунктом 3.1.15 Методики.

Если при разработке ПС используются несколько средств разработки, то это программное средство разбивается на функции таким образом, чтобы каждая функция была реализована с использованием одного средства разработки. Различные функции одного ПС могут быть реализованы с использованием различных средств разработки.

Если в процессе разработки программного средства появляется необходимость реализовать дополнительно одну или несколько функций (задач), то следует оценивать величину трудоемкости доработок следующим образом:

- рассчитать величину трудоемкости  $T_2$  с учетом реализации дополнительных функций (задач) для всего программного средства;
- найти величину трудоемкости доработок, которая будет составлять  $\Delta T = T_2 - T_1$ , где  $T_1$  - величина трудоемкости разрабатываемого ПС без дополнительных функций (задач).

В случае, когда программное средство уже было разработано, и возникает необходимость реализовать дополнительно одну или несколько функций (задач), трудоемкость доработок (развития) рекомендуется рассчитывать одним из следующих способов:

1.1.1. Рассчитать величину трудоемкости разработки  $T_1$  для исходного ПС так, как если бы оно разрабатывалось заново, или в качестве  $T_1$  использовать ранее рассчитанную величину трудоемкости, если для данного ПС такое вычисление проводилось. Затем выполнить все шаги:

- рассчитать величину трудоемкости  $T_2$  с учетом реализации дополнительных функций (задач) для всего программного средства;
- найти величину трудоемкости доработок, которая будет составлять  $\Delta T = T_2 - T_1$ , где  $T_1$  - величина трудоемкости разрабатываемого ПС без дополнительных функций (задач).

1) Выполнить следующие шаги:

- а) оценить трудоемкость необходимых доработок в соответствии с разделом 3.2 Методики, получив величину  $T_3$ . При этом в расчет включаются только те функции (задачи), которые необходимо реализовать дополнительно;
- б) так как Методика предполагает оценку трудоемкости самостоятельного и полнофункционального ПС, а включаемые в расчет функции (задачи) образуют

дорабатываемый функционал ПС, полученный результат необходимо скорректировать по формуле:  $\Delta T = K \cdot T_3$ . Значение коэффициента  $K$  выбирается по следующей таблице:

Отношение объема доработок к объему ПС, $r$ , %	Значение $K$
$r < 10$	0,2
$10 \leq r < 30$	0,4
$30 \leq r < 60$	0,5
$60 \leq r < 90$	0,6
$R \geq 90$	0,7

### 5.3.ПРАВИЛА ПРИМЕНЕНИЯ КОЭФФИЦИЕНТОВ

Настоящий раздел регламентирует правила выбора уровней поправочных коэффициентов для модели расчета трудоемкости разработки ПС согласно порядку расчета, представленному в Методике.

Значение коэффициента сложности функции ( $K_{\text{сложн}}$ ) при расчете объема ПС (пункт 3.1.4 Методики) выбирается согласно уровню сложности функции соответствующего типа операций.

Каждая функция из Каталога соответствует одному из пяти типов операций:

- *управляющие операции;*
- *вычислительные операции;*
- *операции, зависящие от аппаратуры;*
- *операции управления данными;*
- *операции управления пользовательского интерфейса.*

Для определения уровня сложности функции необходимо выполнить следующие шаги:

- 1 шаг. Определить, к какому типу операции относится функция. Тип операции определяется по Каталогу функций (таблица 1.1 Приложения 1 Методики).
- 2 шаг. Определить уровень сложности функции по одной из таблиц 2.1–2.5 Приложения 2 Методики в зависимости от типа операции.
- 3 шаг. Найденному номеру уровня  $L_v$  сопоставить значение из таблицы 2.6 Приложения 2 Методики.

Коэффициент, учитывающий конкретные условия и средства разработки ПС ( $K_{\text{ср.разр}}$ ) при расчете объема ПС (пункт 3.1.4 Методики) выбирается в соответствии с используемым для реализации функции средством разработки.

В случае если необходимое средство разработки в приведенной таблице отсутствует, то, согласно Методике, необходимо самостоятельно отнести его к какому либо средству, наиболее близкому по уровню используемого языка. Аналогично следует поступать при пересчете объема кода согласно пунктам 3.2.1 и 3.2.2 Методики для различных средств разработки с применением переводных коэффициентов (таблица 1.2 Приложения 1 Методики).

Значения коэффициентов, учитывающих уровень требований к показателям качества ПС (требования к надежности ( $K_{\text{над}}$ ), производительности ПС ( $K_{\text{произв}}$ ) и уровню информативности документации ( $K_{\text{докум}}$ ) на этапах жизненного цикла) при расчете величины базовой трудоемкости разработки ПС (пункт 3.2.6 Методики) выбирается с учетом требований заказчика к ожидаемому качеству программного средства.

Допустим, если выполняется разработка АРМ, то уровень требования к надежности можно установить средним, поскольку сбои в работе подобного ПС не приведут к серьезным

последствиям. Требования к надежности выше среднего устанавливаются для ПС, результатом неправильной работы которых может являться возможная потеря важной информации и т.п.

Уровень коэффициента, учитывающего степень новизны ПС ( $K_n$ ) при расчете базовой трудоемкости разработки ПС (пункт 3.1.6 Методики), выбирается в соответствии с таблицей 3.1 Приложения 3 Методики.

Программное средство является принципиально новым в случае, если не имеется доступных аналогов данного ПС.

Программное средство является развитием параметрического ряда ПС в следующих случаях:

- если выполняется разработка новой версии уже существующего программного средства;
- если осуществляется доработка комплекса задач, задачи, функций и создаются новые компоненты, программные модули ПС;

Учет новизны типа ТС и ОС осуществляется следующим образом: если тип ТС/ОС, на котором предполагается функционирование ПС, не использовался в подразделениях Банка России, то он считается новым.

Тип ТС и ОС устанавливается в ТЗ. Выбор значения коэффициента должен соответствовать требованиям ТЗ.

Значение коэффициента, учитывающего использование технологий разработки ( $K_{\text{тех}}$ ) при расчете величины базовой трудоемкости разработки ПС (пункты 3.1.6–3.1.7 Методики) устанавливается в соответствии со структурой ПС и используемыми технологиями взаимодействия компонентов ПС.

Структура ПС и набор технологий взаимодействия компонентов устанавливаются в соответствии с техническим заданием на разработку программного средства. Технологии взаимодействия компонентов могут выбираться для любой структуры программного средства, кроме структуры «Монолитное ПС». Одна или несколько технологий взаимодействия компонентов выбираются, если они реализуются в программном средстве.

Уровни значений коэффициента, учитывающего используемые средства проектирования ( $K_{\text{ср.пр}}$ ) при расчете трудоемкости подпроцесса «Проектирование» (пункт 3.1.10 Методики), соответствуют классам применяемых средств проектирования. Разработчик должен отнести используемые им средства проектирования к одной из предлагаемых групп, наиболее близких по своим возможностям для применяемых средств.

Аналогично устанавливается значение коэффициента, учитывающего используемые среды разработки ( $K_{\text{ср}}$ ) при расчете трудоемкости подпроцесса «Программирование» (пункт 3.1.11 Методики).

Коэффициент, учитывающий используемые средства тестирования ( $K_{\text{ср.тс}}$ ) при расчете трудоемкости подпроцесса «Тестирование» (пункт 3.1.12 Методики), отражает влияние на значение трудоемкости наличия средств автоматизированного тестирования. Значение коэффициента устанавливается в соответствии с описаниями его уровней.

Поправочный коэффициент, учитывающий размер тестовой БД ( $K_{\text{ср.тс}}$ ) при расчете трудоемкости подпроцесса «Тестирование» (пункт 3.1.12 Методики), отражает дополнительные трудозатраты на сбор данных для завершения процесса тестирования. Значение этого коэффициента определяется отношением размера (объема) БД в байтах к количеству строк кода отдельных ПС или комплекса ПС.

Объем данных в БД рассчитывается исходя из размера одной записи и количества записей. Размер одной записи рассчитывается как сумма размеров полей, входящих в запись. Например, тестовая БД состоит из одной таблицы; запись таблицы тестовой БД состоит из двух полей. Первое поле типа INT (размер поля 4 байта), второе – типа VARCHAR (размер

поля 256 байтов). В таком случае размер одной записи составит 260 байтов. Допустим, таблица тестовой БД содержит 100 записей, тогда размер тестовой БД составит  $(260 \cdot 100) = 26000$ .

Значение коэффициента, учитывающего опыт работы программистов со средством разработки ( $K_{\text{опыт}}$ ) при расчете объема ПС (пункт 3.1.4 Методики), выбирается с учетом числа разработанных ПС (компонентов АС) коллективом разработчиков с использованием данного средства разработки по аналогичной тематике. Оценка проводится для всего коллектива разработчиков, участвующего в разработке ПС.

Аналогично устанавливается значение коэффициента, учитывающего опыт разработки программных средств подобного типа ( $K_{\text{ор}}$ ), при расчете базовой трудоемкости разработки ПС (пункт 3.1.6 Методики).

Значение коэффициента, учитывающего опыт работы аналитиков в данной предметной области ( $K_{\text{опыт.ан}}$ ), при расчете трудоемкости подпроцесса «Анализ требований к ПС» (пункт 3.1.9 Методики) и значение коэффициента, учитывающего опыт проектировщиков в данной предметной области ( $K_{\text{опыт.пр}}$ ), при расчете трудоемкости подпроцесса «Проектирование» (пункт 3.1.10 Методики) выбирается с учетом числа разработанных ПС (компонентов АС), относящихся к данной предметной области.

Оценку уровня квалификации персонала, задействованного в выполнении отдельных подпроцессов, следует производить для всего коллектива, занятого в этом подпроцессе. Значение коэффициента, учитывающего уровень квалификации аналитиков ( $K_{\text{квал.ан}}$ ) при расчете трудоемкости подпроцесса «Анализ требований к ПС» (пункт 3.2.9 Методики) устанавливается согласно квалификации персонала.

Аналогично устанавливается значение коэффициента, учитывающего уровень квалификации проектировщиков в данной предметной области ( $K_{\text{квал.пр}}$ ), при расчете трудоемкости подпроцесса «Проектирование» (пункт 3.1.10 Методики), значение коэффициента, учитывающего уровень квалификации программистов ( $K_{\text{квал.прог}}$ ), при расчете трудоемкости подпроцесса «Программирование» (пункт 3.1.11 Методики), значение коэффициента, учитывающего уровень квалификации персонала, осуществляющего тестирование ( $K_{\text{квал.тест}}$ ), при расчете трудоемкости подпроцесса «Тестирование» (пункт 3.1.12 Методики), значение коэффициента, учитывающего уровень квалификации персонала, осуществляющего ввод в действие ПС ( $K_{\text{квал.вн}}$ ), при расчете трудоемкости подпроцесса «Ввод в действие» (пункт 3.1.13 Методики).

Оставляя право формирования коллектива разработчиков, участвующих в разработке ПС для Банка России, за руководством компании-разработчика, рекомендуется относить коллектив разработчиков к имеющим очень высокий уровень опыта в случаях, когда компания сотрудничает с Банком России более 4-х лет и имеет более 5-ти разработанных ПС (компонентов АС) для Банка России или по аналогичной тематике. Кроме того, если компания-разработчик разрабатывает (развивает) отдельные компоненты к ранее разработанному ПС (АС), то уровень квалификации коллектива разработчиков считается как минимум высоким.

#### **5.4. ОСОБЕННОСТИ РАСЧЕТА ТРУДОЕМКОСТИ РАЗРАБОТКИ МАКЕТА ПРОГРАММНОГО СРЕДСТВА**

Данная методика может применяться для расчета макета программного средства. Под макетом понимается совокупность программных элементов, отражающая основные функциональные возможности разрабатываемого программного средства. Это означает, что принцип расчета этапов, модели и дискретный уровень коэффициентов остаются теми же

самыми, что и для расчета ПС, однако выбор качественных и количественных характеристик для макета отличается следующим образом:

– требования к производительности – коэффициент устанавливается на уровень «Требования к производительности ПС не установлены (однако производительность ПС должна обеспечивать приемлемое время отклика при работе пользователя в интерактивном режиме)»;

– информативность документации – коэффициент устанавливается на низкий уровень – «Не учтены некоторые потребности жизненного цикла»;

Функциональный состав макета состоит из базовых функций, которые позволяют решить поставленные в ТЗ задачи, т.е. макет не обязан реализовывать все функции как полноценное ПС. Большая часть внимания, как правило, уделяется алгоритмическим функциям, нежели интерфейсным и транспортным. Кроме того, для функций, не являющихся основными для данного ПС (реализация интерфейса и т.п.), рекомендуется выбирать минимальное значение объема.

## ПРИЛОЖЕНИЕ 6

### ПРИМЕР РАСЧЕТА ТРУДОЕМКОСТИ РАЗРАБОТКИ ПС С ИСПОЛЬЗОВАНИЕМ КАТАЛОГА ФУНКЦИЙ

#### 6.1. Функциональный состав ПС

Определяем функциональный состав программного средства исходя из технического задания на разработку ПС:

1) Компонент «АРМ оператора»:

- добавление, получение и удаление информации о клиенте Банка в базе данных;
- выполнение запросов к базе данных для получения состояния счета, снятия наличных средств со счета, перевода денежных средств клиента со счета в Банке при оплате услуг или перевода денежных средств на счет клиента Банка;

2) Компонент «АРМ администратора»:

- добавление, получение и удаление информации о банкомате;
- добавление, получение и удаление информации об операторе сотовой связи;
- выполнение запросов к базе данных для ввода информации о пополнении денежных средств банкоматов.

3) Компонент «База данных»

- формирование структуры базы данных;
- обеспечение безопасности передачи данных при работе с БД.

4) Компонент «Анализ»:

- проведение анализа о выполненных операциях для прогнозирования необходимого количества наличных денег для банкоматов и сроков их пополнения;
- формирование отчета по анализу.

3) Компонент «Отчет»:

- формирование отчета о выполненных операциях.

#### 6.2. Расчет трудоемкости разработки ПС

Определяем трудоемкость разработки и сроки разработки программного средства. Ниже представлен отчет по расчету трудоемкости.

#### Расчет трудоемкости разработки ПС (Каталог)

##### Параметры расчета

Параметр	Значение
Имя расчета	Программное средство мониторинга финансовых операций, выполняемых через банкомат
Тип расчета	Расчет трудоемкости разработки ПС (Каталог)
Версия модели	2.5.1.0
Фонд времени разработчика [дней/мес.]	21
Тип ограничения	По продолжительности этапа [мес.]
Тип структуры программного средства	ПС с набором библиотек (=1.00)
Технологии взаимодействия модулей	Оригинальный протокол взаимодействия (sockets) (=0.80)

**Коэффициенты уровня расчета**

Коэффициент	Уровень значения
Степень новизны ПС	Принципиально новое ПС, не имеющее доступных аналогов на прежнем типе ТС/ОС (=1.06)
Требования к надежности	Сбои ПС приводят к потерям рабочей информации (=1.06)
Требования к производительности	Повышенные требования к производительности (=1.12)
Информативность документации	Повышенный объем для жизненного цикла данного ПС (=1.07)
Опыт разработки ПС подобного типа	Более 5 разработанных ПС (компонентов АС) (=0.81)
Влияние сроков работ на трудоемкость	Равно или более 100 (=1.00)

**Коэффициенты уровня подпроцесса**

Подпроцесс	Коэффициент	Уровень значения
Анализ требований	Уровень квалификации аналитиков	Высокий (=0.87)
Анализ требований	Опыт работы аналитиков в данной предметной области	5 разработанных ПС (компонентов АС) (=0.87)
Проектирование	Уровень квалификации проектировщиков	Высокий (=0.92)
Проектирование	Опыт работы проектировщиков с используемыми средствами проектирования	5 разработанных ПС (компонентов АС) (=0.92)
Проектирование	Используемые средства проектирования	Средства проектирования БД (BPWIN, ERWIN/ERX, Designer 2000, DataDirect Explorer, Oracle Designer, Silverrun RBM и BPM, Together J) (=1.03)
Программирование	Уровень квалификации программистов	Высокий (=0.95)
Программирование	Используемая среда разработки	Интегрированные среды разработки (Visual Studio) (=1.00)
Тестирование	Уровень квалификации персонала, осуществляющего тестирование	Высокий (=0.89)
Тестирование	Используемые средства тестирования	Автоматизированные средства тестирования использовались (=0.76)
Тестирование	Размер БД	10 <= D/P < 100 (=1.00)
Ввод в действие ПС	Уровень квалификации персонала, осуществляющего ввод в действие ПС	Высокий (=0.94)

**Дерево элементов**

№	Элемент	Описание	Vi (на условном языке)	Кср.разр.	ri	ki [0;1]	Ксложн.	Копыт
	АРМ Оператора	Компонент предназначен для работы оператора с программным средством						
2.2.5	+Корректировка балансов**	Реализует интерфейс оператора ПС, обеспечивающий работу оператора с программным средством.	500 из [200-500]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
2.1.15	+Ввод прочих данных	Добавление информации о клиенте Банка в БД программного средства.	3000 из [1000-4000]	C++ (=1.00)	1	0.85	Высокий (=1.10)	Высокий (=0.91)
1.5.12	+Поиск, представление и вывод информации	Получение и удаление информации о клиенте Банка в базе данных, выполнение запросов к базе данных для получения состояния счета, снятия наличных средств со счета, перевода денежных средств клиента со счета в Банке при оплате услуг или перевода денежных сред	3500 из [1000-6000]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
	АРМ Администратора	Компонент предназначен для работы администратора с программным средством						
2.2.5	+Корректировка балансов**	Реализует интерфейс администратора ПС, обеспечивающий работу администратора с программным средством.	500 из [200-500]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
2.1.15	+Ввод прочих данных	Добавление информации о банкомате, об операторе сотовой связи, выполнение запросов к базе данных для ввода информации о пополнении денежных средств банкоматов.	3000 из [1000-4000]	C++ (=1.00)	1	0.85	Высокий (=1.10)	Высокий (=0.91)
1.5.12	+Поиск, представление и вывод информации	Получение и удаление информации о банкомате и операторе сотовой связи.	3500 из [1000-6000]	C++ (=1.00)	1	0.90	Высокий (=1.10)	Высокий (=0.91)
	База Данных	Компонент предназначен для хранения информации о выданных кредитных картах, операций по снятию наличных, оплаты услуг сотовой связи.						

1.5.1	+Формирование физической структуры аналитических БД (на одну БД)	Обеспечивает формирование структуры базы данных (для обеспечения ее нормального функционирования с разрабатываемым программным средством).	4500 из [1000-8000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
2.1.1	+Ввод данных первичных документов в интерактивном режиме	Обеспечивает безопасность передачи информации из базы данных (согласно требованиям технического задания на разработку ПС).	2000 из [2000-9000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
	Отчет	Компонент предназначен для составления отчета о выполняемых операциях (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета).						
4.6.1	+Формирование отчетов (на один отчет)*	Формирование отчета о выполненных операциях (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета).	2050 из [100-4000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
	Анализ	Компонент предназначен для прогнозирования сроков пополнения и количества необходимых наличных денег для банкоматов.						
5.1.2	+Настройка аналитических таблиц	Проведение анализа о выполненных операциях (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета) для прогнозирования сроков пополнения и количества необходимых наличных денег для банкоматов.	1000 из [1000-2000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
4.6.1	+Формирование отчетов (на один отчет)*	Формирование отчета по анализу выполненных операций (снятие наличных со счета, оплата услуг сотовой связи, получение информации о состоянии счета) через банкоматы Банка.	2050 из [100-4000]	C++ (=1.00)	1	1.00	Высокий (=1.10)	Высокий (=0.91)
	Общий объем ПС:		22010					

**Расчет трудоемкости подпроцессов**

<b>Подпроцесс</b>	<b>Трудоемкость [чел.дн.]</b>	<b>Кол-во разработчиков [чел]</b>	<b>Продолжительность [мес]/[дн]</b>
Анализ требований	4.89	0.12	2.00/42.00
Проектирование	67.55	1.07	3.00/63.00
Программирование	484.60	5.77	4.00/84.00
Тестирование	30.57	0.73	2.00/42.00
Ввод в действие ПС	6.07	0.29	1.00/21.00
Всего	593.69		12.00/252.00
Базовая трудоемкость	645.71		