



Областное государственное образовательное
учреждение среднего профессионального
образования «Иркутский авиационный
техникум»

«УТВЕРЖДАЮ»
Директор
ОГБОУ СПО "ИАТ"
Семенов В.Г. Семенов
«31» августа 2012 г.

**Фонд оценочных средств
по профессиональному модулю**

ПМ.03 Участие в интеграции программных модулей
образовательной программы
по специальности СПО

09.02.03 Программирование в компьютерных системах

г.Иркутск

Рассмотрена
цик洛вой комиссией

Протокол № _____
от «____» ____ 20__ г.

Председатель ЦК
_____ / _____ /

№	Разработчик ФИО (полностью)
1	Петкевич Любовь Викторовна

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Область применения фонда оценочных средств (ФОС)

ФОС профессионального модуля – является частью образовательной программы в соответствии с ФГОС СПО по специальности

09.02.03 Программирование в компьютерных системах

в части освоения вида профессиональной деятельности (ВПД):

Участие в интеграции программных модулей

и соответствующих профессиональных компетенций (ПК):

ПК.3.1 Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

ПК.3.2 Выполнять интеграцию модулей в программную систему.

ПК.3.3 Выполнять отладку программного продукта с использованием специализированных программных средств.

ПК.3.4 Осуществлять разработку тестовых наборов и тестовых сценариев.

ПК.3.5 Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.

ПК.3.6 Разрабатывать технологическую документацию.

1.2 Цели и задачи модуля – требования к результатам освоения модуля

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен:

В результате освоения дисциплины обучающийся должен	№ дидактической единицы	Формируемая дидактическая единица
Знать	1.1	модели процесса разработки программного обеспечения;
	1.2	основные принципы процесса разработки программного обеспечения;
	1.3	основные подходы к интегрированию программных модулей;
	1.4	основные методы и средства эффективной разработки;
	1.5	основы верификации и аттестации программного обеспечения;

	1.6	концепции и реализации программных процессов;
	1.7	принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения;
	1.8	методы организации работы в коллективах разработчиков программного обеспечения;
	1.9	основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов;
	1.10	стандарты качества программного обеспечения;
	1.11	методы и средства разработки программной документации
Уметь	2.1	владеть основными методологиями процессов разработки программного обеспечения;
	2.2	использовать методы для получения кода с заданной функциональностью и степенью качества;
Иметь практический опыт	3.1	участия в выработке требований к программному обеспечению;
	3.2	участия в проектировании программного обеспечения с использованием специализированных программных пакетов;

2. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МЕЖДИСЦИПЛИНАРНЫХ КУРСОВ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ ИСПОЛЬЗУЕМЫЙ НА ТЕКУЩЕМ КОНТРОЛЕ

2.1 Результаты освоения МДК.03.01 подлежащие проверке на текущем контроле

№ текущего контроля	Индекс занятия текущего контроля	Проверяемая дидактическая единица	Формируемые компетенции	Основные показатели оценивания результата	№ задания относящийся к показателю оценивания	Метод контроля	Форма контроля	Вид контроля	Индексы занятий ранее изученных связанные с контролируемыми единицами
1	1.2.8	1.1	ПК.3.1	1.1.1.1	1.1	Опрос	Практическая работа	Анализ выполнения задания	1.1.2
		1.2	ПК.3.1	1.1.2.1	1.1	Опрос	Практическая работа	Анализ выполнения задания	1.1.1, 1.2.1, 1.2.3
		1.3	ПК.3.1	1.1.3.1	1.1	Опрос	Практическая работа	Анализ выполнения задания	1.2.2
		1.4	ПК.3.1	1.1.4.1, 1.1.4.2	1.1, 1.1	Опрос	Практическая работа	Анализ результатов выполнения задания	1.2.4, 1.2.5, 1.2.6, 1.2.7
2	1.2.14	1.1	ПК.3.1,	2.1.1.1	2.1	Опрос	Практическая	Анализ	

		ПК.3.6				работа	выполнения задания	
	1.2	ПК.3.1, ПК.3.6	2.1.2.1	2.1	Опрос	Практическая работа	Анализ выполнения задания	1.2.9
	1.3	ПК.3.1, ПК.3.6	2.1.3.1	2.1	Опрос	Практическая работа	Анализ выполнения задания	
	1.4	ПК.3.1, ПК.3.6	2.1.4.1, 2.1.4.2, 2.1.4.3	2.1, 2.2, 2.2	Опрос	Практическая работа	Анализ результатов выполнения задания	1.2.8, 1.2.10, 1.2.11, 1.2.12, 1.2.13
3	1.3.13	1.1	ПК.3.1, ПК.3.6, ПК.3.2	3.1.1.1	3.1	Опрос	Практическая работа	Анализ выполнения задания
		1.2	ПК.3.1, ПК.3.6, ПК.3.2	3.1.2.1, 3.1.2.2	3.1, 3.3	Опрос	Практическая работа	Анализ выполнения задания
		1.3	ПК.3.1, ПК.3.6, ПК.3.2	3.1.3.1	3.3	Опрос	Практическая работа	Анализ выполнения задания
		1.4	ПК.3.1, ПК.3.6, ПК.3.2	3.1.4.1	3.3	Опрос	Практическая работа	Анализ выполнения задания

									1.3.10, 1.3.11, 1.3.12
4	1.4.4	1.1	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4	4.1.1.1	4.1	Опрос	Практическая работа	Анализ выполнения задания	
		1.2	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4	4.1.2.1, 4.1.2.2, 4.1.2.3, 4.1.2.4, 4.1.2.5	4.1, 4.2, 4.3, 4.4, 4.5	Опрос	Практическая работа	Анализ выполнения задания	1.3.13, 1.4.1
		1.3	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4	4.1.3.1	4.6	Опрос	Практическая работа	Анализ выполнения задания	
		1.4	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4	4.1.4.1	4.6	Опрос	Практическая работа	Анализ выполнения задания	1.3.14, 1.3.15, 1.3.16, 1.3.17, 1.3.18, 1.4.2, 1.4.3
5	2.4.3	1.1	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4, ПК.3.5	5.1.1.1	5.1	Опрос	Индивидуальн ые задания	Анализ выполнения этапа курсового прое ктирования	
		1.2	ПК.3.1, ПК.3.6,	5.1.2.1	5.1	Опрос	Индивидуальн ые задания	Анализ выполнения	

		ПК.3.2, ПК.3.4, ПК.3.5				этапа курсового прое- ктирования	
1.3	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4, ПК.3.5	5.1.3.1	5.1	Опрос	Индивидуальн- ые задания	Анализ выполнения этапа курсового прое- ктирования	
1.4	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4, ПК.3.5	5.1.4.1	5.1	Опрос	Индивидуальн- ые задания	Анализ выполнения этапа курсового прое- ктирования	1.4.4
1.6	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4, ПК.3.5	5.1.6.1	5.1	Опрос	Индивидуальн- ые задания	Анализ выполнения этапа курсового прое- ктирования	1.5.1, 1.5.2
2.1	ПК.3.1, ПК.3.6, ПК.3.2, ПК.3.4, ПК.3.5	5.2.1.1	5.1	Опрос	Индивидуальн- ые задания	Анализ выполнения этапа курсового прое- ктирования	2.1.1, 2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.2.1, 2.2.2, 2.4.1
2.2	ПК.3.1, ПК.3.6, ПК.3.2,	5.2.2.1	5.1	Опрос	Индивидуальн- ые задания	Анализ выполнения этапа	2.3.1, 2.3.2, 2.4.2

			ПК.3.4, ПК.3.5				курсового прое ктирования
--	--	--	-------------------	--	--	--	------------------------------

Перечень заданий текущего контроля

Номер задания	Задания
1.1	<p>Проектирование структур данных</p> <p>Цель работы: Рассмотреть процесс проектирования структур данных разрабатываемого программного обеспечения.</p> <ol style="list-style-type: none"> 1. Ознакомиться с теоретическим материалом по проектированию структур данных. 2. Рассмотреть предложенные в учебнике примеры по проектированию структур данных. 3. Разработать структуру данных по предложенной информационной системе. 4. Оформить отчет по работе.
2.1	Диаграмма последовательностей системы

Цель работы: Рассмотреть процесс описания поведения разрабатываемого программного обеспечения при объектном подходе.

1. Ознакомиться с теоретическим материалом по анализу программного обеспечения, основанного на использовании диаграмм последовательностей системы.
2. Рассмотреть предложенные в учебнике примеры по составлению диаграмм последовательностей системы.
3. Построить диаграмму последовательностей системы для проектируемого программного обеспечения.
4. Оформить отчет по работе.

2.2

Диаграмма деятельности

Цель работы: Рассмотреть процесс описания поведения разрабатываемого программного обеспечения при объектном подходе.

1. Ознакомиться с теоретическим материалом по анализу программного обеспечения, основанного на использовании диаграмм деятельности.
2. Рассмотреть предложенные в учебнике примеры по составлению диаграммы деятельности.
3. Построить диаграмму деятельности для проектируемого программного обеспечения.
4. Оформить отчет по работе.

3.1

Основные компоненты пользовательских интерфейсов

Цель работы: Ознакомиться с основными компонентами графического пользовательского интерфейса.

1. Ознакомиться с теоретическим материалом разработки пользовательского интерфейса.
2. Рассмотреть предложенные в учебнике примеры по применению компонентов пользовательского интерфейса.
3. Произвести выбор компонентов графического пользовательского интерфейса для проектируемого программного обеспечения.
4. Оформить отчет по работе.

3.3	<p>Проектирование пользовательского интерфейса прямого манипулирования</p> <p>Цель работы: Проектирование пользовательского интерфейса прямого манипулирования.</p> <ol style="list-style-type: none"> 1. Ознакомиться с теоретическим материалом по проектированию пользовательского интерфейса прямого манипулирования. 2. Рассмотреть предложенные в учебнике примеры по проектированию пользовательского интерфейса прямого манипулирования. 3. Разработать пользовательский интерфейс прямого манипулирования для проектируемого программного обеспечения. 4. Оформить отчет по работе.
4.1	<p>Ручной контроль программного обеспечения</p> <p>Цель работы: Проектирование этапов контроля программного обеспечения.</p>

	<ol style="list-style-type: none"> 1. Ознакомиться с теоретическим материалом по применению ручного контроля программного обеспечения. 2. Рассмотреть предложенные в учебнике примеры по проектированию этапов ручного контроля программного обеспечения. 3. Разработать список вопросов при анализе правильности программного обеспечения индивидуального задания. 4. Оформить отчет по работе.
4.2	<p>Структурное тестирование</p> <p>Цель работы: Тестирование программного обеспечения с помощью структурного тестирования.</p> <ol style="list-style-type: none"> 1. Ознакомиться с теоретическим материалом по применению структурного тестирования программного обеспечения. 2. Рассмотреть предложенные в учебнике примеры по применению структурного тестирования. 3. Разработать список маршрутов тестирования программного обеспечения индивидуального задания. 4. Оформить отчет по работе.
4.3	<p>Функциональное тестирование</p>

Цель работы: Ознакомиться с методикой функционального тестирования.

1. Ознакомиться с теоретическим материалом по применению функционального тестирования программного обеспечения.
2. Рассмотреть предложенные в учебнике примеры по использованию функционального тестирования.
3. Разработать методику тестирования программного обеспечения индивидуального задания.
4. Оформить отчет по работе.

4.4

Комплексные методы тестирования

Цель работы: Ознакомиться с методами комплексного тестирования.

1. Ознакомиться с теоретическим материалом по применению комплексного тестирования программного обеспечения.
2. Рассмотреть предложенные в учебнике примеры по использованию комплексного тестирования.

	<ol style="list-style-type: none">3. Разработать рекомендации по комплексному тестированию программного обеспечения индивидуального задания.4. Оформить отчет по работе.
4.5	<p>Оценочное тестирования</p> <p>Цель работы: Ознакомиться с методом оценочного тестирования.</p> <ol style="list-style-type: none">1. Ознакомиться с теоретическим материалом по применению оценочного тестирования программного обеспечения.2. Разработать рекомендации по оценочному тестированию программного обеспечения индивидуального задания.3. Оформить отчет по работе.
4.6	<p>Методы отладки программного обеспечения</p>

Цель работы: Ознакомиться с методами отладки программного обеспечения.

1. Ознакомиться с теоретическим материалом по использованию методов отладки программного обеспечения.
2. Разработать рекомендации по методам отладки программного обеспечения индивидуального задания.
3. Оформить отчет по работе.

5.1

Стадии выполнения курсового проекта

1. Формирование требований к АИС
2. Разработка концепции АИС
3. Техническое задание
4. Эскизный проект
5. Технический проект
6. Рабочая документация
7. Ввод в действие

8. Сопровождение АИС

Перечень показателей текущего контроля

Номер показателя	Значение показателя
1.1.1.1	Понимание стадий жизненного цикла программ
1.1.2.1	Понимание процесса разработки программ
1.1.3.1	Понимание методов разработки структуры программ
1.1.4.1	Описание структуры данных
1.1.4.2	Соответствие описания структуры данных заданию
2.1.1.1	Понимание процесса разработки программ
2.1.2.1	Понимание процесса разработки программ
2.1.3.1	Понимание процесса разработки программ
2.1.4.1	Способность разрабатывать диаграмму последовательностей
2.1.4.2	Способность разрабатывать диаграмму деятельности
2.1.4.3	Соответствие разработанных диаграмм последовательностей и деятельности
3.1.1.1	Понимание процесса разработки программ
3.1.2.1	Способность производить выбор компонентов пользовательского интерфейса
3.1.2.2	Способность реализовывать диалоги пользовательского интерфейса

3.1.3.1	Понимание процесса разработки интерфейса
3.1.4.1	Понимание процесса разработки интерфейса
4.1.1.1	Понимание процесса разработки программ
4.1.2.1	Понимание метода ручного тестирования
4.1.2.2	Понимание метода структурного тестирования
4.1.2.3	Умение метода структурного тестирования
4.1.2.4	Понимание метода комплексного тестирования
4.1.2.5	Понимание метода оценочного тестирования
4.1.3.1	Понимание методов отладки программ
4.1.4.1	Способность определять методы тестирования и отладки программного обеспечения
5.1.1.1	Понимание методов определения требований к программному обеспечению
5.1.2.1	Понимание процесса разработки программного обеспечения
5.1.3.1	Понимание методов интеграции программных модулей
5.1.4.1	Понимание методов эффективной разработки
5.1.6.1	Понимание методов реализации программных продуктов
5.2.1.1	Способность применять методологию процесса разработки программного обеспечения
5.2.2.1	Способность получать код программных продуктов в соответствии с заданным алгоритмом

2.2 Результаты освоения МДК.03.02 подлежащие проверке на текущем контроле

№ текущего ко	Индекс занятия текущим	Проверяемая дидактика	Формируемые компетенции	Основные показатели оценивания	№ задания относящийся	Метод контроля	Форма контроля	Вид контроля	Индексы занятий ранее изученных

нтра ля	щего к онтрол я	ическая я един ица		ния резул ьтата	ся к показ ателю оце нивания				связанные с ко нтролируемым и дидактическ ими единицами
1	2.2.5	1.2	ПК.3.1, ПК.3.2, ПК.3.3	1.1.2.1, 1.1.2.2, 1.1.2.3, 1.1.2.4, 1.1.2.5	1.1, 1.1, 1.1, 1.2, 1.2	Опрос	Практическая работа	Анализ выполнения задания	1.1.1, 1.1.2, 1.1.3
		1.4	ПК.3.1, ПК.3.2, ПК.3.3	1.1.4.1	1.2	Опрос	Практическая работа	Анализ выполнения задания	2.1.1, 2.1.2, 2.1.3, 2.1.4, 2.2.1, 2.2.2, 2.2.3, 2.2.4
2	2.2.15	1.2	ПК.3.1, ПК.3.2, ПК.3.3, ПК.3.6	2.1.2.1	2.6	Опрос	Самостоятельн ая работа	Анализ выполнения задания	
		1.4	ПК.3.1, ПК.3.2, ПК.3.3, ПК.3.6	2.1.4.1, 2.1.4.2, 2.1.4.3, 2.1.4.4, 2.1.4.5	2.1, 2.2, 2.3, 2.4, 2.5	Опрос	Практическая работа	Анализ выполнения задания	2.2.5, 2.2.6, 2.2.7, 2.2.8, 2.2.9, 2.2.10, 2.2.11, 2.2.12, 2.2.13, 2.2.14
		1.7	ПК.3.1, ПК.3.2, ПК.3.3,	2.1.7.1	2.5	Опрос	Практическая работа	Анализ выполнения задания	2.2.5

			ПК.3.6					
		2.2	ПК.3.1, ПК.3.2, ПК.3.3, ПК.3.6	2.2.2.1	2.5	Опрос	Практическая работа	Анализ выполнения задания
3	2.2.21	1.2	ПК.3.1, ПК.3.2, ПК.3.3, ПК.3.6, ПК.3.4	3.1.2.1	3.3	Опрос	Самостоятельн ая работа	Анализ выполнения задания
		1.4	ПК.3.1, ПК.3.2, ПК.3.3, ПК.3.6, ПК.3.4	3.1.4.1, 3.1.4.2, 3.1.4.3, 3.1.4.4, 3.1.4.5	3.1, 3.1, 3.1, 3.2, 3.2	Опрос	Практическая работа	Анализ выполнения задания
		1.7	ПК.3.1, ПК.3.2, ПК.3.3, ПК.3.6, ПК.3.4	3.1.7.1	3.1	Опрос	Практическая работа	Анализ выполнения задания
		2.2	ПК.3.1, ПК.3.2, ПК.3.3, ПК.3.6, ПК.3.4	3.2.2.1	3.2	Опрос	Практическая работа	Анализ выполнения задания
								2.2.5

Перечень заданий текущего контроля

Номер задания	Задания
1.1	<p>Создание контекстной диаграммы для данной задачи.</p> <p>Цель работы:</p> <ol style="list-style-type: none">1. Знакомство с примером2. Создание контекстной диаграммы для данного примера. <p>Постановка задачи. Пример</p> <p>В качестве примера рассмотрим деятельность вымышленной компании «Олди», которая существует 5 лет и занимается в основном сборкой и продажей настольных компьютеров и ноутбуков. Годовой оборот компании составляет примерно 20 млн. долларов. Компания закупает компоненты для компьютеров от трех независимых поставщиков, а не производит компоненты самостоятельно.. она только собирает и тестирует компьютеры. Компания реализует продукцию через магазины и специализируется на покупателях, для которых главный критерий при покупке – стоимость компьютера. Предполагаемый объем рынка для компании «Олди» в последующие 2 года – 50 млн.долл.</p> <p>Несмотря на некоторое увеличение объема продаж, прибыли уменьшаются, растет конкуренция на рынке.</p>

Чтобы не потерять позиции компания решает проанализировать текущие бизнес-процессы и реорганизовать их с целью увеличения эффективности производства и продаж. Основные процедуры в компании таковы:

- Продавцы принимают заказы клиентов;
- Операторы группируют заказы по типам компьютеров;
- Операторы собирают и тестируют компьютеры;
- Операторы упаковывают компьютеры согласно заказам;
- Кладовщик отгружает клиентам заказы.

В настоящее время компания «Олди» использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

Улучшение деятельности компании должно касаться структур управления компанией, эффективности производства и внутреннего контроля. В результате реорганизация может потребовать внедрения новой корпоративной информационной системы (состоящей не только из одного бухгалтерского модуля).

Однако перед тем как пытаться производить какие-то улучшения необходимо разобраться в существующих бизнес-процессах.

Создание контекстной диаграммы.

Для составления контекстной диаграммы выполните следующие действия.

1. Запустите BPwin. Кнопка Пуск/Программы/
2. Появится диалоговое окно ModelMart Connection Manager. Нажмите кнопку Cancel.
3. Появится диалоговое окно Iwould like to. Впишите имя модели Деятельность компании «Олди» и выберите Type- IDEF0. Нажмите кнопку OK.
4. Появится окно Properties for New Models, в окно Autor впишите свою фамилию и нажмите кнопку OK.
5. Автоматически создается контекстная диаграмма.
6. Обратите внимание на кнопку на панели инструментов. Эта кнопка включает и выключает инструмент просмотра и навигации - Model Explorer (появляется слева). Кнопка Activities/Diagram переключает режим Model Explorer. В режиме Activities щелчок правой кнопкой мыши по объекту в Model Explorer позволяет редактировать его свойства.
7. Перейдите в меню **Model/Model Properties**. В закладке General диалогового окна Model Properties введите имя модели {Деятельность компании «Олди»}, имя проекта (Project) {Модель деятельности «Олди»}, Имя автора свою фамилию и тип модели Time Frame{AS-IS}.
8. В закладке **Purpose** внесите Цель { Purpose: Моделировать текущие {AS-IS} бизнес-процессы компании «Олди» } и точку зрения {Viewpoint: Директор}.
9. В закладке **Definition** внесите определение {Это учебная модель, описывающая деятельность

компании «Олди» и Scope(область действия) {Общее управление бизнесом компании: исследование рынка, закупка компьютеров, сборка, тестирование и продажа продуктов}.

10. В закладке **Source** (источник информации) введите {Материалы курса по BPwin}
11. В закладке **Status** установите WORKING и нажмите кнопку OK.
12. Перейдите на контекстную диаграмму и правой кнопкой мыши щелкните по работе. В контекстном меню выберите Name . В закладке Name внесите имя {Деятельность компании «Олди»}.
13. В закладке **Definition** внесите определение {Текущие бизнес-процессы компании «Олди»}.
14. В закладке **Status** установите WORKING.
15. В закладке **Source** внесите {Материалы курса по BPwin } и нажмите кнопку OK .

Создайте стрелки на контекстной диаграмме (табл.1.), для создания стрелок прочитайте «Стрелки» и «Границные стрелки».

Таблица 1.

Наименование стрелки (Arrow Name)

Описание (Arrow Definition)

Тип

Бухгалтерская система

Оформление счетов, оплата счетов работы с заказами

Механизм

Звонки клиентов

Запросы информации, заказы, техническая поддержка и т.д.

Вход

Правила и процедуры

Правила продажи, инструкции по сборке, процедуры тестирования, критерии производительности и т.д.

Управление

Проданные продукты

Настольные и портативные компьютеры

Выход

16. Для создания **Arrow Name**, установите курсор на соответствующую стрелку, правой кнопкой мыши вызовите контекстное меню, выберите пункт Name и впишите название стрелки, например, Бухгалтерская система.

17. Перейдите на вкладку **Definition** и внесите описание для соответствующей стрелки, например, Оформление счетов, оплата счетов, работа с заказами.
18. Создайте отчет по модели, используя меню **Tools/Reports/Model Reports**
19. Сохраните свою работу в своей папке.

Контрольные вопросы:

1. Как начать работу в программе BPwin?.
2. Как внести имя модели ?
3. Как внести имя проекта?
4. Как внести имя автора?
5. Как внести цель моделирования?
6. Как установить свойства диаграммы?
7. Как сохранить модель?
8. Как установить шрифт объекта?

	9. Как установить цвет объекта?
	10. Как создать отчет по модели?
	11. Показать на диаграмме, где находится тип стрелки «Механизм»?
	12. Показать на диаграмме, где находится тип стрелки «Вход»?
	13. Показать на диаграмме, где находится тип стрелки «Управление»?
	14. Показать на диаграмме, где находится тип стрелки «Выход»?

1.2	<p>Создание диаграммы декомпозиции.</p> <p><i>Цель работы:</i></p> <p>1. Создание диаграммы декомпозиции для данного примера.</p> <p>Создание диаграммы декомпозиции.</p> <p>Перед выполнением этой работы внимательно прочтайте раздел «Работа» и раздел «Стрелки»</p> <p>1. Выберите кнопку перехода на нижний уровень в палитре инструментов, в диалоговом окне Activity Box Count установите число работ 3 на диаграмме нижнего уровня нажмите кнопку OK</p>
-----	---

Автоматически будет создана диаграмма декомпозиции. Правой кнопкой мыши щелкните по первой работе, выберите Name и внесите имя работы, например, «Продажи и маркетинг». Повторите операцию для всех трех работ. Затем внесите определение, статус и источник для каждой работы.

Описание работ для диаграммы декомпозиции

Функциональный блок Name)

Описание Definition)

Статус (Status)

Источник (Source)

Продажи, маркетинг

Телемаркетинг, презентации, выставки

WORKING

Материалы курса по BPwin

Сборка, тестирование компьютеров

Сборка и тестирование настольных и портативных компьютеров

WORKING

Материалы курса по BPwin

Отгрузка, получение

Отгрузка заказов клиентам и получение компонентов от поставщиков

WORKING

Материалы курса по BPwin

2. Для изменения свойств работ после их внесения в диаграмму можно воспользоваться словарем объектов модели. Вызов словаря **Model/Diagram Object Editor**

3. Если Вы опишите имя и свойства работы в словаре, её можно будет внести в диаграмму позже с помощью кнопки в палитре инструментов. Вы не можете удалить работу из словаря, если она используется на какой-либо диаграмме. Если вы удалите работу из диаграммы, из словаря она не удаляется. Имя и описание такой работы может быть использовано в дальнейшем. Для добавления работы в словарь щелкните по кнопке **Clear**, внесите имя и свойства работы, затем щелкните по **Add**. Для удаления всех работ, не использующихся в модели, щелкните по **Purge**

3. Перейдите в режим рисования стрелок. Свяжите граничные стрелки (кнопка на палитре инструментов) с остальными. Для этого прочтайте раздел «Разветвляющиеся и сливающиеся стрелки».

4. Правой кнопкой мыши щелкните по ветви стрелки управления работы «Сборка и тестирование компьютеров» и переименуйте её в «Правила сборки и тестирования».

Внесите определение (Definition) для новой ветви: «Инструкция по сборке, процедуры тестирования, критерии производительности и т.д.»

Правой кнопкой мыши щелкните по ветви стрелки механизма работы «Продажи и маркетинг» и переименуйте её в «Систему оформления заказов»

5. Альтернативный метод внесения имен и свойств стрелок – использование словаря стрелок (вызов словаря – меню Model/Diagram Object editor). Если вы опишите имя и свойство стрелки в словаре, её можно будет внести в диаграмму позже. Вы не можете удалить стрелку из словаря, если она используется на какой-либо диаграмме. Если Вы удалите стрелку из диаграммы, из словаря она не удаляется. Имя и описание такой стрелки может быть использовано в дальнейшем. Для добавления стрелки в словарь щелкните по кнопке Clear, внесите имя и свойства работы, затем щелкните по Add. Для удаления всех имен стрелок, не использующихся в модели, щелкните по Purge Unused.

6. Создайте новые внутренние стрелки.

7. Создайте стрелку обратной связи (по управлению) «Результаты сборки и тестирования», идущую от работы «Сборка и тестирование компьютеров» к работе «Продажи и маркетинг». Для большей наглядности измените стиль стрелки (толщина линий) и установите опцию Extra Arrowhead (из контекстного меню). Методом drag&drop перенесите имена стрелок так, чтобы их было удобнее читать. Если необходимо, установите Squiggle (из контекстного меню).

8. Создайте новую граничную стрелку выхода «Маркетинговые материалы» из работы «Продажи и маркетинг». Эта стрелка автоматически не попадает на диаграмму верхнего уровня и имеет квадратные

скобки на наконечнике. Из палитры выберите кнопку, щелкните мышью по квадратным скобкам и в диалоговом окне Border Editor выберите Resolve Border Arrow. Для стрелки «Маркетинговые материалы» выберите опцию Trim из контекстного меню.

Контрольные вопросы:

1. Как создать стрелку входа?
2. Как создать стрелку выхода ?
3. Как создать стрелку управления?
4. Как создать стрелку механизма?
5. Как посмотреть Словарь стрелок?
6. Как создать внутренние стрелки?
7. Как создать стрелки обратной связи?
8. Как создать разветвляющиеся и сливающиеся стрелки?
9. Как создать диаграмму декомпозиции?
10. Как создать отчет по модели?

2.6

2.2 Стадии разработки ПО, регламентированных ГОСТами

В нашей стране жизненный цикл разработки ПО установлен стандартом ГОСТ 19.102-77 Стадии разработки программ и программной документации и содержит следующие стадии и этапы:

- 1. Техническое задание (ТЗ).**
- 2. Эскизный проект (ЭП).**
- 3. Технический проект (ТП).**
- 4. Рабочий проект (РП).**
- 5. Внедрение.**

В табл. 2.1 показаны стадии разработки и этапы, их составляющие.

Неверно предполагать, что жизненный цикл разработки ПО согласно ГОСТ 19.102-77 есть последовательное выполнение стадий и этапов, определенных в таблице 2. В реальном жизненном цикле трудно провести четкую и определенную границу между этапами, а сам процесс создания ПО является итеративным: после завершения некоторого этапа почти всегда есть необходимость в коррекции уже выполненных этапов и стадий с целью внесения уточнений. При разработке принципиально нового ПО иногда бывает необходимо осуществить пробную реализацию с целью получения информации, требующейся для принятия решения на некоторой стадии.

Таблица 2.1

Стадии разработки

Этапы работ

Техническое задание

1. Обоснование необходимости разработки программ.
2. Выполнение научно-исследовательских работ (НИР).
3. Разработка и утверждение технического задания.

Эскизный проект

1. Разработка эскизного проекта.
2. Утверждение эскизного проекта.

Технический проект

1. Разработка технического проекта.
2. Утверждение технического проекта.

Рабочий проект

1. Разработка программы.
2. Разработка программной документации.
3. Испытание программы.

Внедрение

1. Подготовка и передача программы.

Специалистам в области разработки ПО известно, что наиболее важными стадиями в жизненном цикле разработки являются начальные, так как ошибки, допущенные на них, требуют значительных затрат на исправление на конечных стадиях.

Техническое задание. На стадии Техническое задание выполняются следующие работы, входящие в состав соответствующих этапов.

1. Обоснование необходимости разработки программ:

постановка задачи;

сбор исходных материалов;

выбор и обоснование критериев эффективности и качества;

обоснование необходимости проведения НИР.

2. Выполнение научно-исследовательских работ:

определение структуры входных и выходных данных;
предварительный выбор методов решения задач;
обоснование целесообразности применения ранее разработанных программ;
определение требований к техническим средствам;
обоснование принципиальной возможности решения поставленных задач.

3. Разработка и утверждение технического задания:

определение требований к программе;
разработка технико-экономического обоснования разработки программы;
определение стадий, этапов и сроков разработки программы и документации на нее;
выбор языков программирования;
определение необходимости проведения НИР на последующих стадиях;
согласование и утверждение ТЗ.

Результатом выполнения данной стадии является **техническое задание**, оформленное в соответствии с ГОСТ

19.105-78 (изм. 09.1981.) Общие требования к программным документам и ГОСТ 19.106-78 Общие требования к программным документам, выполненным печатным способом на листах формата 11 и 12 (по ГОСТ 2.301-68).

Эскизный проект. Основные этапы и содержание работ на стадии Эскизный проект приведены в таблице 2.2.

Таблица 2.2

Этапы работ

Содержание

Разработка ЭП

1. Предварительная разработка структуры входных и выходных данных.
2. Уточнение методов решения задач.
3. Разработка общего описания алгоритма решения задачи.
4. Разработка технико-экономического обоснования.

Утверждение ЭП

1. Разработка пояснительной записки.

2. 2. Согласование и утверждение эскизного проекта.

Конкретное содержание работ стадии эскизного проекта и их объем определяет степень сложности разрабатываемого ПО. Результатом выполнения данной стадии является полное описание архитектуры ПО. Как правило, это описание делается на нескольких уровнях иерархии. На верхнем уровне детализации выделяются основные подсистемы, которым присваиваются имена, устанавливаются связи между подсистемами, их функции, получаемые путем декомпозиции предполагаемых функций ПО. Затем процедура декомпозиции выполняется для каждой подсистемы, выделяются модули, составляющие данную подсистему. В конечном итоге, получается иерархически организованная система, состоящая из уровней, каждый из которых представляет собой совокупность взаимосвязанных модулей.

Единицы, выделяемые на различных иерархических уровнях функциональной архитектуры системы, определяются по усмотрению разработчика. Стандарты ЕСПД различают программные единицы только с точки зрения их документирования.

Результаты эскизного проекта отображаются в документе Пояснительная записка к эскизному проекту, оформленному в соответствии с ГОСТ 19.105-78 и ГОСТ 19.404-79.

После утверждения пояснительной записи она становится программным документом, правила дублирования, учета, хранения которого определяется ГОСТ 19.601-78 Общие правила дублирования, обращения, учета и хранения и ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. Последующие стадии и этапы разработки ПО могут выявить необходимость внесения изменений в ЭП. Эти изменения должны быть отражены в пояснительной записке в соответствии с ГОСТ 19.603-78 Общие правила внесения изменений в программные документы и ГОСТ 19.602-78 Правила внесения изменений в программные документы, выполненные печатным способом.

В качестве примера, ниже приводится фрагмент расширенного описания работ стадии эскизного проекта.

1. Разработка эскизного проекта ПО.

разработка плана совместных работ на разработку ПО;

разработка и обоснование математической модели системы на ЭВМ и описание результатов моделирования;

разработка и обоснование алгоритмов и временных графиков функционирования ПО по всем режимам работы;

разработка и обоснование ресурсов памяти для реализации алгоритмов;

разработка перечня документов на ПО;

разработка и обоснование структуры БД, внешних входных и выходных данных;

разработка и обоснование алгоритмов информационного обеспечения;

определение взаимосвязей между видами программ;

разработка и обоснование набора тестов для проверки ПО;

разработка и обоснование организации наращивания и развития ПО;

оформление пояснительной записки и ведомости эскизного проекта ПО (в соответствии с ГОСТ 19.105-78, ГОСТ 19.404-79 и ГОСТ 2.106-68 ЕСКД. Текстовые документы);

согласование и утверждение ЭП.

Технический проект. Основные этапы и содержание работ на стадии Технический проект приведены в таблице 2.3.

Содержанием работ на этой стадии является проектирование структуры ПО. Результатом - реализующий заданный и утвержденный в техническом задании комплекс программ как иерархическая структура программных модулей, заданных своими функциональными спецификациями. Форма представления результата - Пояснительная записка к техническому проекту согласно ГОСТ 19.105-78, ГОСТ 19.404-79.

Разработка структуры ПО заключается в выделении всех программных компонентов по функциональным признакам, определение функциональных спецификаций модулей и уточнение внешних функциональных спецификаций, структуры входных и выходных данных, определении операционной среды, языковых средств и конфигурации аппаратных средств.

Спецификации модулей являются внешними характеристиками и содержат все сведения, необходимые вызывающим модулям. На последующих стадиях разработки спецификации оформляются в виде комментариев в начале текста исходной программы модуля. На данной стадии спецификации оформляются в виде комментария на принятом в организации, занимающейся разработкой ПО, языке спецификаций

Таблица 2.3.

Этапы работ

Содержание

Разработка ТП

1. Уточнение структуры входных и выходных данных.
2. Разработка алгоритмов решения задач.
3. Определение формы представления входных и выходных данных.
4. Определение синтаксиса и семантики языка.
5. Разработка структуры программы.
6. Окончательное определение конфигурации технических средств.

Утверждение ТП

1. Разработка плана мероприятий по разработке и внедрению программ.
2. Разработка пояснительной записки.
3. Согласование и утверждение ТП.

Фрагмент описания работ стадии технического проекта приведен ниже.

1. Разработка технического проекта ПО.

разработка плана совместных работ на разработку ПО;

распараллеливание задач применительно к возможностям ЭВМ и систем обмена информации и разработка организационной структуры программ;

формализация программно-аппаратных и внутризадачных интерфейсов;

разработка алгоритмов управления вычислительным процессом и процессами обмена, его исследование и оптимизация;

разработка и комплексное решение вопросов обеспечения устойчивой работы на программном уровне;

разработка модели и исследование на ней возможности реализации всех функций системы управления в рамках имеющихся вычислительных ресурсов на основе выбранного управляющего алгоритма и оптимизация структуры программ до получения удовлетворяющего результата;

уточнение структуры и определение формы представления БД, внешних входных и выходных данных;

разработка проекта описания программы для каждого вида программ ПО;

уточнение алгоритмов ПО и алгоритмов информационного обеспечения в процессе решения задач;

разработка проекта описания текстовых программ ПО;

разработка плана реализации ПО;

разработка проектов технических условий на ПО;

разработка и обоснование решений по надежности функционирования ПО;

разработка программы и правил испытаний ПО;

уточнение временных графиков функционирования программ ПО в процессе решения задач;

оформление пояснительной записи и ведомости технического проекта ПО (в соответствии с ГОСТ 19.105-78, ГОСТ 19.404-79 и ГОСТ 2.106-68 ЕСКД. Текстовые документы);

согласование и утверждение ТП.

Рабочий проект. Основные этапы и содержание работ на стадии Рабочий проект приведены в таблице 2.4.

Таблица 2.4

Этапы работ

Содержание

Разработка ПО

- Программирование и отладка программ.

Разработка программной документации

1. Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

Испытание ПО

1. Разработка, согласование и утверждение программ и методики испытаний
2. Проведение предварительных государственных, межведомственных приемо-сдаточных и других видов испытаний.
3. Корректировка ПО и программной документации по результатам испытаний.

Содержанием работ на этой стадии является описание ПО на выбранном проблемно-ориентированном языке (кодирование), отладка, разработка, согласование и утверждение порядка и методики испытаний, разработка программных документов, проведение тестирования, корректировка программ и программной документации по результатам тестирования, проведение приемо-сдаточных испытаний. Результат - ПО в форме программной документации, в форме документации на ПО или в форме программного изделия.

На стадии **Внедрения** осуществляется подготовка и передача ПО и программной документации для сопровождения и/или изготовления, оформление и утверждение акта о передаче ПО на сопровождение или

изготовление, передача ПО в фонд алгоритмов и программ.

Кроме рассмотренного выше жизненного цикла программ, установленного 19 - ой системой стандартов, необходимо иметь в виду, что существует жизненный цикл автоматизированных систем, установленный ГОСТ 34.601-90. Имеет смысл рассмотреть стадии и этапы разработки АС, так как 19-я система стандартов в настоящее время морально устарела и более современным представляется жизненный цикл АС. Анализ жизненных циклов разработки ПО, установленных ГОСТ 19.102 и ГОСТ 34.601, а также международных стандартов, регламентирующих данный процесс, будет приведен ниже.

Стандарт ГОСТ 34.601 распространяется на автоматизированные системы (АС), представляющие собой организационно-технические системы, обеспечивающую выработку решений на основе автоматизации информационных потоков в различных видах деятельности (исследование, проектирование, управление и т.п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях. Данный стандарт устанавливает стадии и этапы создания АС.

В процессе разработки АС создают, в общем случае, следующие виды обеспечения: техническое, программное, информационное математическое и др. Проектные решения по программному, техническому и информационному обеспечению реализуют как изделия в виде взаимоувязанной совокупности компонент и комплексов, входящей в состав АС с необходимой документацией. Справочное приложение к ГОСТ 34.601 устанавливает, что программное обеспечение АС - совокупность программ на носителях информации с программной документацией по ГОСТ 19.101. Таким образом, при разработке программного обеспечения можно пользоваться как стандартом ГОСТ 19.102, так и ГОСТ 34.601.

Процесс создания АС представляет собой совокупность упорядоченных во времени, взаимосвязанных, объединенных в стадии и этапы работ, выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям. Стадии и этапы создания АС выделяются как части процесса создания по соображениям рационального планирования и организации работ, заканчивающих заданным

результатом.

Как и в стандарте ГОСТ 19.102, работы по развитию АС осуществляются по стадиям и этапам, применяемым для создания АС. Состав и правила выполнения работ на установленных настоящим стандартом стадиях и этапах определяют в соответствующей документации организаций, участвующих в создании конкретных видов АС.

Стадии и этапы создания АС в общем случае приведены в таблице 2.5.

Стандартом ГОСТ 34.601 допускается исключать стадию Эскизный проект и отдельные этапы работ на всех стадиях, объединять стадии Технический проект и Рабочая документация в одну стадию Технорабочий проект. В зависимости от специфики создаваемых АС и условий их создания допускается выполнять отдельные этапы работ до завершения предшествующих стадий, параллельное во времени выполнения этапов работ, включение новых этапов работ.

Рассмотрим этапы, касающиеся разработки программного обеспечения.

При Обследование объекта и обоснование необходимости создания АС проводят: сбор данных об объекте автоматизации и осуществляемых видах деятельности; оценку качества функционирования объекта и осуществляемых видов деятельности, выявления проблем, решение которых возможно средствами автоматизации; оценку (технико-экономической, социальной и т.п.) целесообразности создания АС.

На этапе Формирование требований пользователя к АС проводят:

подготовку исходных данных для формирования требований к АС (характеристика объекта автоматизации, описание требований к системе, ограничения допустимых затрат на разработку, ввод в действие и эксплуатацию, эффект, ожидаемый от системы, условия создания и функционирования);

формулировку и оформление требований пользователя к АС.

Таблица 2.5.

Стадии

Этапы работ

1. Формирование требований к АС

1.1. Обследование объекта и обоснование необходимости создания АС

1.2. Формирование требований пользователя к АС

1.3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)

2. Разработка концепции АС

2.1. Изучение объекта

2.2. Проведение необходимых научно - исследовательских работ

2.3. Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя

2.4. Оформление отчета о выполненной работе

3. Техническое задание

3.1. Разработка и утверждение технического задания на создание АС

4. Эскизный проект

4.1. Разработка предварительных проектных решений по системе и ее частям

4.2. Разработка документации на АС и ее части

5. Технический проект

5.1. Разработка проектных решений по системе и ее частям

5.2. Разработка документации на АС и ее части

5.3. Разработка и оформление документации на поставку изделий для комплектования АС и/или технических требований (технических заданий) на их разработку

5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации

6. Рабочая документация

6.1. Разработка рабочей документации на систему и ее части

6.2. Разработка или адаптация программ

7. Ввод в действие

7.1. Подготовка объекта автоматизации к вводу АС в действие

7.2. Подготовка персонала

7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)

7.4. Строительно-монтажные работы

7.5. Пусконаладочные работы

7.6. Проведение предварительных испытаний

7.7. Проведение опытной эксплуатации

7.8. Проведение приемочных испытаний

8. Сопровождение АС

8.1. Выполнение работ в соответствии с гарантийными обязательствами

8.2. Послегарантийное обслуживание

На этапе Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания) проводят оформление отчета о выполненных работах на данной стадии и оформление заявки на разработку АС (тактико-технического задания) или другого заменяющего ее документа с аналогичным содержанием.

При Изучение объекта и Проведение необходимых НИР организация-разработчик проводит детальное изучение объекта автоматизации и необходимые НИР, связанные с поиском путей и оценкой возможности реализации требований пользователя, оформляют и утверждают отчеты о НИР.

Этап Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя направлен на разработку альтернативных вариантов концепции создаваемой АС и планов реализации; оценку необходимых ресурсов на их реализацию и обеспечение функционирования; оценку преимуществ и недостатков каждого варианта; сопоставление требований пользователя и характеристик предлагаемой системы и выбор оптимального варианта; определение порядка оценки качества и условий приемки системы; оценку эффектов, получаемых от системы.

На этапе Оформление отчета о выполненной работе подготавливают и оформляют отчет, содержащий описание выполненных работ на стадии, описание и обоснование предлагаемого варианта концепции системы.

На этапе Разработка и утверждение технического задания на создание АС проводят разработку, оформление, согласование и утверждение технического задания на АС и, при необходимости, технических заданий на части АС.

Результатом выполнения этапа Разработка предварительных проектных решений по системе и ее частям

является: функции АС; функции подсистем, их цели и эффекты; состав комплекса задач и отдельных задач; концепции информационной базы, ее укрупненная структура; функции СУБД; состав вычислительной системы; функции и параметры основных программных средств.

На этапе Разработка проектных решений по системе и ее частям обеспечивают разработку общих решений: по системе и ее частям, функционально-алгоритмической структуре системы; по функциям персонала и организационной структуре; по структуре технических средств; по алгоритмам решения задач и применяемым языкам; по организации и ведению информационной базы, системе классификации и кодирования информации; по программному обеспечению.

На этапах 4.2 и 5.2 Разработка документации на АС и ее части проводят разработку, оформление, согласование и утверждение документации в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию АС.

На этапе Разработка и оформление документации на поставку изделий для комплектации АС и (или) технических требований (технических заданий) на их разработку проводят: подготовку и оформление документации на поставку изделий для комплектования АС; определение технических требований и составление ТЗ на разработку изделий, не изготавливаемых серийно.

На этапе Разработка заданий на проектирование в смежных частях проекта объекта автоматизации осуществляют разработку, оформление, согласование и утверждение заданий на проектирование в смежных частях проекта объекта автоматизации для проведения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием АС.

При выполнении этапа Разработка рабочей документации на систему и ее части осуществляют разработку рабочей документации, содержащей все необходимые и достаточные сведения для обеспечения выполнения работ по вводу АС в действие и ее эксплуатации, а также для поддерживания уровня эксплуатационных

характеристик (качества) системы в соответствии с принятыми проектными решениями, ее оформление, согласование и утверждение. Виды документов - по ГОСТ 34.201.

На этапе Разработка и адаптация программ проводят разработку программ и программных средств системы, выбор, адаптацию и (или) привязку приобретенных программных средств, разработку программной документации в соответствии с ГОСТ 19.101.

На этапе Подготовка объекта автоматизации к вводу АС в действие проводят работы по организационной подготовке объекта автоматизации к вводу АС в действие, в том числе: реализацию проектных решений по организационной структуре АС; обеспечение подразделений объекта управления инструктивно-методическими материалами; внедрение классификаторов информации.

Этап Подготовка персонала направлен на обучение персонала и проверку его способности обеспечить функционирование АС.

На этапе Комплектация АС поставляемыми изделиями обеспечивают получение комплектующих изделий серийного и единичного производства, материалов и монтажных изделий. Проводят входной контроль их качества.

На этапе Пусконаладочные работы проводят автономную наладку технических и программных средств, загрузку информации в БД и проверку системы ее ведения; комплексную наладку всех средств системы.

На этапе Проведение предварительных испытаний осуществляют: испытание АС на работоспособность и соответствие ТЗ в соответствии с программой и методикой предварительных испытаний; устранение неисправностей и внесение изменений в документацию на АС, в том числе эксплуатационную в соответствии с протоколом испытаний; оформление акта о приемке АС в опытную эксплуатацию.

На этапе Проведение опытной эксплуатации проводят: опытную эксплуатацию АС; анализ результатов опытной эксплуатации АС; доработку (при необходимости) программного обеспечения АС; дополнительную наладку (при необходимости) технических средств АС; оформление акта о завершении опытной эксплуатации.

На этапе Проведение приемочных испытаний проводят испытания на соответствие ТЗ в соответствии с программой и методикой приемочных испытаний, анализ результатов испытаний АС и устранение недостатков, выявленных при испытаниях, а также оформление акта о приемке АС в постоянную эксплуатацию.

На этапе Выполнение работ в соответствии с гарантийными обязательствами осуществляются работы по устранению недостатков, выявленных при эксплуатации АС в течение установленных гарантийных сроков, внесению необходимых изменений в документацию на АС.

На этапе Послегарантийное обслуживание осуществляются работы по анализу функционирования системы, выявлению отклонений фактических эксплуатационных характеристик АС от проектных значений, установлению причин этих отклонений, устранению выявленных недостатков и обеспечение стабильности эксплуатационных характеристик АС, внесению необходимых изменений в документацию на АС.

Требования к содержанию документов, разрабатываемых на каждой стадии и этапе, устанавливает руководящий документ РД-50-34.698-90. Кроме того, необходимо использовать для из разработки соответствующие стандарты Единой системы программной документации, Единой системы конструкторской документации и ГОСТ 34.602. Виды и комплектность документов регламентированы в ГОСТ 34.201.

2.1

Создание сущностей и атрибутов на диаграмме

Цель работы: Изучение процесса создания объектов модели данных сущностей и атрибутов, используя панель инструментов; также процесса создания новых, переименования, удаления атрибутов в сущности; освоения навыков задания свойств сущностям и атрибутам.

Основные понятия работы - сущность и атрибут

Создание модели данных в ERwin, как правило, начинается с создания логической модели данных.

В ERwin сначала создается сущность, и ей задается имя, потом в ней создаются атрибуты, и одному или нескольким из них при необходимости задается параметр Primary Key (первичный ключ).

Задания:

1. Создать структуру модели данных, состоящую из следующих сущностей: Заказ, Заказчик, Дизайн и Реализация.
2. Для каждой сущности соответственно создать следующие атрибуты:

Номер заказа. Дата заказа, Наименование продукции, Количество для сущности Заказ.

Номер заказчика. Фамилия, Имя, Отчество, Адрес, Телефон, Статус заказчика для сущности Заказчик.

Номер чертежа. Размер, Цвет, Стиль для сущности Дизайн.

Номер реализации. Дата реализации, Наименование продукции. Количество, Стоимость для сущности Реализация.

Задать свойство Primary Key всем первым атрибутам в списках выше.

1. Для каждой сущности задать свойство Definition по своему усмотрению.
2. Для атрибутов задать свойства Definition и Domain по своему усмотрению.

Технология работы

1. **Создание сущности.** Для того, чтобы начать создание сущности необходимо открыть или создать файл проекта ERwin в меню File, стандартным методом. Затем, нажав на кнопку сущности на панели инструментов щелкнуть в любом месте диаграммы, затем задать имя сущности.
2. **Создание атрибутов.** После создания сущности, нужно создать в ней атрибут, для этого необходимо правой кнопкой мыши щелкнуть на сущности и в контекстном меню выбрать Attributes... После этого, в правой части появившегося диалога, снизу, нажать на кнопку New; и задать имя нужного атрибута. Для редактирования имени существует кнопка Rename, для удаления - Delete. Для задания уже созданному атрибуту свойства Primary Key. Для отображения первичного ключа на диаграмме, нужно в не занятом диаграммами месте щелкнуть правой кнопкой мыши и в entity display выделить Primary Key Designator.
3. **Задание свойств сущностям и атрибутам.** Для задания свойства Definition сущности нужно щелкнув по

сущности правой кнопкой и в контекстном меню выбрать пункт Entity Properties, в появившемся диалоге выбрать закладку Definition и ввести нужный текст. Для задания аналогичного свойства атрибуту необходимо также щелкнуть по сущности выбрав пункт Attributes, и в появившемся диалоге выбрать закладку Definition. Для задания типа данных атрибуту нужно выбрать в том же диалоговом окне вкладку General и выбрать нужный домен для атрибута. Для смены стандартной иконки для сущности, во вкладке Icon нужно щелкнуть по раскрывающемуся списку и выбрать иконку, если список пуст, то, используя кнопку Import можно импортировать любой тип иконки. Для отображения иконки сущности на диаграмме, нужно в не занятом диаграммами месте щелкнуть правой кнопкой мыши и в Entity Display выделить Entity Icon.

Контрольные вопросы

1. В чём назначение стандарта IDEF1?
2. Основные понятия нотации IDEF1.
3. Какие уровни представления и уровни отображения имеет ERwin? Как происходит переключение между уровнями?
4. Каким образом осуществляется взаимосвязь между двумя отдельными сущностями?
5. Что называется моделью данных и концептуальной схемой?
6. Что описывает сущность в IDEF1X и в чём её отличие от сущности в IDEF1?
7. Назначение атрибутов сущностей в IDEF1X.

8. Как графически описывается сущность в диаграмме IDEF1X?
9. Как идентифицировать уникальным образом запись сущности?
10. Какие правила существуют для выбора первичного ключа?
11. Как классифицируются сущности в IDEF1X?

2.2

Создание связей между сущностями

Цель работы: Изучение процесса создания объектов модели данных - связей, используя панель инструментов, также освоения навыков задания свойств связям.

Основное понятие работы - связь

Связь является логическим соотношением между сущностями. Каждая связь должна именоваться глаголом или глагольной фразой (Relationship Verb Phrases). Имя связи выражает некоторое ограничение или бизнес-правило и облегчает чтение диаграммы. ERwin использует два типа связей идентифицирующая и неидентифицирующая. Идентифицирующая связь устанавливается между независимой (родительский конец связи) и зависимой (дочерний конец связи) сущностями. Когда рисуется идентифицирующая связь, ERwin автоматически преобразует дочернюю сущность в зависимую. Зависимая сущность изображается прямоугольником со скругленными углами. При установлении идентифицирующей связи атрибуты

первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности.

При установлении неидентифицирующей связи дочерняя сущность остается независимой, а атрибуты первичного ключа родительской сущности мигрируют в состав неключевых компонентов родительской сущности. Неидентифицирующая связь служит для связывания независимых сущностей.

Задания:

1. Создать идентифицирующую связь между сущностями Заказчик и Заказ, Заказ и Дизайн. Чтобы сущность Заказ была дочерней для сущности Заказчик, а для сущности Дизайн - родительской.
2. Создать неидентифицирующую связь между сущностями Заказ и Реализация. Чтобы сущность Заказ была родительской для сущности Реализация.
3. Задать свойство *Verb Phrase* для всех связей по принципу Заказчик размещает Заказ, Дизайн описывает Заказ, Заказ обеспечивает Реализацию.

Технология работы

1. **Создание связи.** Для создания связи между сущностями нужно нажать соответствующую кнопку на панели инструментов (в зависимости от типа нужной связи), щелкнуть на родительской сущности, затем на дочерней.

2. Задание свойства Verb Phrase. Для задания свойства Verb Phrase нужно вызвать диалог Relationship, щелкнув по связи правой кнопкой и выбрав пункт Relationship Properties.. Затем выбрать вкладку General и вписать в поле Parent-to-Chaild необходимый глагол или глагольную фразу. Для отображения свойства Verb Phrase на диаграмме, нужно в не занятом диаграммами месте щелкнуть правой кнопкой мыши и в Relationship Display выделить Verb Phrase.

Контрольные вопросы

1. Для чего устанавливается связь между сущностями в IDEF1X?
2. Какие типы связей существуют между сущностями?
3. Как графически отличить зависимую и независимую сущности, идентифицирующие и неидентифицирующие связи?
4. Для чего и как задаются свойства сущностям и атрибутам?
5. Для чего предназначен диалог Relationship?

2.3

Экспорт модели данных *ERwin* в модель процессов *BPwin*

Цель работы: Изучение процесса экспорта модели данных в модель процессов.

При создании КИС модель данных может содержать сотни таблиц, в таком случае возникает необходимость

сравнить модель данных с моделью процессов, для гарантии того, что она не конфликтует с существующими моделями объектов.

ERwin позволяет производить экспорт только на уровне логической модели. Для успешного связывания моделей необходимо, чтобы версии ERwin и BPwin соответствовали друг другу.

Задание:

Экспортировать модель данных в модель процессов, связать сущность и атрибуты с работами и стрелками в модели процессов.

Технология работы

Для экспорта модели данных из ERwin в BPwin необходимо в ERwin открыть модель(проверить, чтобы никакие объекты не были выделены) и выбрать пункт меню File/Export/BPwin.

В появившемся диалоге Select BPwin Export File необходимо выбрать каталог, указать имя создаваемого файла экспорта *.eah и нажать Сохранить.

Затем в BPwin нужно открыть модель процессов, выбрать в меню пункт File/Import/ERwin (EAX), в диалоге Open выбрать имя файла (*.eah) и нажать OK. Появится диалог Import Differences Preview, в котором показывается протокол импорта. Для внесения данных в модель процессов следует щелкнуть по кнопке Accert. Кнопка Cancel отменяет импорт.

После внесения данных в модель процессов можно связать сущности и атрибуты со стрелками. Правой кнопкой мыши нужно щелкнуть по стрелке и выбрать в контекстном меню Arrow Data. Появляется вкладка Arrow Data диалога Arrow Properties. Для связывания атрибута со стрелкой достаточно щелкнуть по иконке выбора в иерархическом списке атрибутов. При этом сущность в модели процессов может быть связана с несколькими атрибутами различных сущностей.

В свою очередь работы тоже могут воздействовать на данные. Для документирования такого воздействия необходимо щелкнуть правой кнопкой мыши по работе и выбрать пункт меню Data Usage Editor.

В появившемся диалоговом окне Data Usage Editor в виде иерархического списка показывается все работы модели, стрелки, которые касаются работ, сущности и атрибуты. Для задания ассоциации достаточно щелкнуть по окну в иерархическом списке.

Для сущностей задается ассоциация CRUD (Create, Read, Update, Delete), для атрибутов - IRUN (Insert, Read, Update, Nullify). Ассоциация CRUD и IRUN - это правила использования сущностей и атрибутов работами, т.е. то, что могут делать работы с входящими и исходящими данными. Данные не могут использоваться работами произвольно.

Для проверки соответствия связи атрибутов и сущностей с работами и стрелками, открываем модель данных и модель процессов, щелкаем по работе в модели процессов появляется диалог Data Usage Editor, в модели данных щелкаем правой кнопкой мыши по соответствующей сущности и из контекстного меню выбираем пункт Attributes, и в диалоговом окне Attributes появляется список атрибутов, которые соответствуют стрелкам в модели процессов.

Контрольные вопросы

1. Провести сравнение созданных моделей ERwin и BPwin на уровне характеристики объектов.
2. Как происходит связывание объектов модели данных со стеклами и работами?
3. Каким образом графически происходит преобразование стрелки в сущность (информация о стрелке в нескольких сущностях) ?
4. Рассказать алгоритм экспорта модели данных из ERwin в BPwin.
5. Для чего предназначен диалог Arrow Properties - вкладка Arrow Data?
6. Для чего предназначено диалоговое окно Data Usage Editor?
7. Как применяются правила использования сущностей и атрибутов работами(ассоциации CRUN и IRUN)?
8. Провести графическое сравнение сущностей и атрибутов со стрелками и работами.

2.4

Вычисление размера БД

Цель работы: Изучение процесса вычисления размера БД с помощью диалога Volumetrics, освоение навыков работы с диалогом Volumetrics.

ERwin позволяют рассчитать приблизительный размер БД в целом, а также таблиц, индексов, и других объектов через определенный период времени, после начала эксплуатации ИС.

Задание:

Вычислить приблизительный размер БД через 24 месяца, если каждый месяц будет выполняться 50 заказов (начальное значение 0), таблица заказчиков будет пополняться на 25 заказчиков ежемесячно (начальное значение 10), таблица реализации будет пополняться на 50 заказов (начальное значение 0), и таблица дизайна будет увеличиваться каждый месяц на 30 (начальное значение 0).

Технология работы

Для расчета размеров физических объектов служит диалог Volumetrics, который вызывается из меню Tools/Volumetrics...

Редактор Volumetrics имеет три закладки - Settings, Report и Parameters.

Settings. Служит для задания основных параметров, на основе которых вычисляется размер БД:

В группе **Table Row Counts** для выделенной в левом списке Table таблицы задается начальное количество строк (Initial), максимальное количество строк (Max) и прирост количества строк в месяц (Grow Bu). Если параметры Max и Grow Bu используются одновременно, рост размера таблицы прекращается по достижении максимального размера. Те же самые параметры можно задать для каждой таблицы в закладке Volumetrics редактора Tables. Сразу после задания параметров Initial, Max и Grow Bu в группе Sizing Estimates, расположенной в левом нижнем углу диалога, показывается средний размер строки, начальный размер таблицы и индексов.

Таблица **Column Properties** позволяет задать свойства колонок таблицы. Имена колонок, их тип и размер

(allocated) не редактируются. Можно изменять ширину поля Avg Width (для тех типов данных, для которых это допускается) и параметр Per NULL (средний ожидаемый процент строк, в которых текущее поле принимает значение NULL). ERwin автоматически определяет в зависимости от выбранной СУБД, какие поля таблицы Column Properties могут изменяться.

Группа **Include Indexes** позволяет учесть или игнорировать индексы, создаваемые на внешних (FK, Foreign Key), первичных (PK, Primary Key), альтернативных (AK, Alternate Key) ключах или инверсионных входах (IE, Inverse Entry) при расчете размера БД.

Группа **Storage** позволяет задать объект физической памяти, в котором будет храниться выбранная таблица. Если объект физической памяти не описан, его можно определить в соответствующем редакторе.

Report. В ней отображаются результаты расчета размера БД (рис. 1.14). Группа Options позволяет выбрать тип объектов, по которым проводится расчет, Time - временной диапазон (начальное состояние или определенное время после начала эксплуатации).

Parameters. Служит для задания дополнительных параметров, используемых для расчета размера БД:

TableFactor. Этот фактор показывает накладные расходы на хранение таблицы в БД. Например, значение TableFactor = 2 увеличит размер таблиц вдвое.

IndexFactor показывает накладные расходы на хранение индекса в БД. Например, значение IndexFactor = 1 увеличит размер индекса с 1 М до 1,5 М .

RowOverhead используется для дополнительного пересчета количества байт каждой строки. Например, если значение RowOverhead 10, размер каждой строки таблицы будет увеличен на 10 байт.

BlobFactor и **BlobBlockSize** используется для пересчета Blob-колонок, хранящихся физически вне таблицы.

BytesPerChar используется для задания количества байт, необходимых для хранения одного символа строкового типа. Для ASCII - это 1, для других кодировок значение может быть больше 1, например для UNICODE - 2.

LogPercent используется для вычисления размеров log-файлов БД. LogPercent = 100 увеличивает БД вдвое.

Контрольные вопросы

1. Как размер БД связан с эксплуатацией ИС?
2. Как задаются основные параметры, на основе которых вычисляется размер БД?
3. Как рассчитать размер БД через определенное время после начала эксплуатации?
4. Как влияют временные таблицы на размер БД?

2.5

Прямое проектирование

Цель работы: Изучение процесса прямого проектирования БД с помощью диалога Forward Engineer; освоение навыков, работая с диалогом Forward Engineer.

Основное понятие работы - прямое проектирование

Процесс генерации физической схемы БД из логической модели данных называется прямым проектированием (Forward Engineering). При генерации физической схемы ERwin включает триггеры ссылочной целостности, хранимые процедуры, индексы, ограничения и другие возможности, доступные при определении таблиц в выбранной СУБД. Процесс генерации логической модели из физической БД называется обратным проектированием (Reverse Engineering). ERwin позволяет создать модель данных путем обратного проектирования имеющейся БД. После того как модель создана, можно переключиться на другой сервер (модель будет конвертирована) : и произвести прямое проектирование структуры БД для другой СУБД. Кроме режима прямого обратного проектирования ERwin поддерживает синхронизацию между логической моделью и системным каталогом СУБД на протяжении всего жизненного цикла создания ИС.

Задание:

1. Создать один пустой файл в СУБД MS Access 2000 и пустую папку в списке Database MS SQL Server 2000 под названием «Rabola» сохранить и закрыть.
2. Открыть файл «l access.erl» путем прямого проектирования сгенерировать код для создания системного каталога БД в файл «Rabota.mdb».
3. Открыть файл «lsql.erl» путем прямого проектирования сгенерирован, код для создания системного каталога БД в список Database в папку под названием Rabota.

Технология работы

Для генерации системного каталога БД следует выбрать пункт меню Tools/Forward Engineer/Schema Generation или нажать кнопку на панели инструментов. Появляется диалог Schema Generation.

Диалог Schema Generation имеет три закладки:

Options. Служит для задания опций генерации объектов БД - триггеров, таблиц, представлений, колонок, индексов и т. д. Для задания опций генерации какого-либо объекта следует выбрать объект в левом списке закладки, после чего включить соответствующую опцию в правом списке.

В закладке **Summary** отображаются все опции, заданные в закладке Options. Список опций в Summary можно редактировать так же, как и в Options.

Comment. Позволяет внести комментарий для каждого набора опций.

Каждый набор опций может быть именован (окно Option Set, кнопки New, Rename и Delete) и использован многократно.

Кнопка **Preview** вызывает диалог Schema Generation Preview, в котором отображается SQL-скрипт, создаваемый ERwin для генерации системного каталога СУБД. Нажатие на кнопку Generate приведет к запуску процесса генерации схемы.

Кнопка **Print** предназначена для вывода на печать создаваемого ERwin SQL-скрипта.

Кнопка **Report** сохраняет тот же скрипт в ERS или SQL текстовом файле. Эти команды можно в дальнейшем редактировать любым текстовым редактором и выполнять при помощи соответствующей утилиты сервера.

Кнопка Generate запускает процесс генерации схемы. Возникает диалог связи с БД, устанавливается сеанс связи с сервером и начинает выполняться SQL-скрипт. При этом возникает диалог Generate Database Schema.

В процессе установки сеанса связи с Access:

Поле User Name - заполняем Admin (сокращение от Administrator);

Поле Database - прописываем путь к базе данных Access, заранее созданной (Rabota.mdb).

Поле System Database - прописываем путь к файлу system.mdw (который может находиться в папке C:\Program Files\Microsoft Office 2000\Office).

Остальные поля не заполняем и нажимаем кнопку Connect.

В процессе установки сеанса связи SQL Server:

В диалоговом окне SQL Server Connection ставим переключатель в строке Use Windows authentication.

В поле Database - заполняем именем заранее созданной БД (Rabota);

Поле Server Name - заполняем именем сервера.

Нажимаем кнопку Connect.

По умолчанию в диалоге Generate Database Schema ключена опция Stop If Failure. Это означает, что при первой же ошибке выполнение скрипта прекращается. Щелкнув по кнопке Continue, можно продолжить выполнение. Кнопка Abort прерывает выполнение. При выключенной опции Stop If Failure скрипт будет

выполняться, несмотря на встречающиеся ошибки.

Дополнительная информация

Можно создать один файл модели данных, физический уровень которого соответствует MS Access 2000, код которого путем прямого проектирования сгенерировать в БД, которую затем заполнить данными.

После этого физический уровень модели конвертировать, т.е. сменить СУБД на MS SQL Server 2000. Имена таблиц и связи модели данных менять нельзя. При смене СУБД ERwin предлагает автоматически преобразовать тип данных, связанный с каждым атрибутом, на ближайший, доступный для новой СУБД. Для автоматического преобразования следует в ответ на вопрос нажать Yes, после этого проверить типы данных для всех атрибутов. Затем путем прямого проектирования сгенерировать код в БД SQL Server:

После этого открываем MS Access 2000, в диалоге Microsoft Access выбираем переключатель Мастера, страницы и проекты баз данных нажимаем Ok, затем в закладке Общие выбираем Проект (существующая база данных) нажимаем Ok. Задаем имя БД (*.adp) нажимаем Ok. После этого появляется диалог Data Link Properties, в строке Select or Enter a server name - задаем имя сервера, после этого в списке Enter Information to Log on the server - ставим переключатель в строке Use Windows NT Integrated security. В раскрывающемся списке Select the database on the server выбираем созданную ранее БД, в которую был сгенерирован код модели, нажимаем Ok. БД связывается с сервером, в результате структура переносится в БД *.adp.

Для того чтобы импортировать данные из заполненной БД *.mdb, нужно открыть базу данных *.adp, из меню Файл выбрать пункт Внешние данные Импорт, затем выбрать БД *.mdb, из которой будем импортировать данные, нажать Ok. На экране появится диалоговое окно Импорт объектов, задаем все объекты БД (таблицы, формы и т.д.) выделяя их в списке, после этого нажимаем Ok, запуская процесс импорта объектов. После окончания процесса импорта все одноименные импортированные таблицы в имени будут содержать либо 1, либо знак подчеркивания. Следующим шагом следует удалить пустые таблицы, а

импортированные переименовать (убрать 1 или знак подчеркивания), чтобы их имена совпадали с именами таблиц хранящихся на сервере. После этого открываем структуру БД в SQL Server, где в таблицах должны отражаться данные из *.adp.

Контрольные вопросы

1. Дать определение-триггера, хранимой процедуры.
2. Что называется триггером ссылочной целостности?
3. Для чего предназначен индекс на физическом уровне?
4. Что называется прямым и обратным проектированием?
5. Как выполнить генерацию системного каталога?
6. Что является результатом прямого проектирования для СУБД (MS Access 2000 и MS SQL Server 2000)?
7. В чем эффективность прямого проектирования?

3.3

2.2 Стадии разработки ПО, регламентированных ГОСТами

В нашей стране жизненный цикл разработки ПО установлен стандартом ГОСТ 19.102-77 Стадии разработки программ и программной документации и содержит следующие стадии и этапы:

- 1. Техническое задание (ТЗ).**
- 2. Эскизный проект (ЭП).**
- 3. Технический проект (ТП).**
- 4. Рабочий проект (РП).**
- 5. Внедрение.**

В табл. 2.1 показаны стадии разработки и этапы, их составляющие.

Неверно предполагать, что жизненный цикл разработки ПО согласно ГОСТ 19.102-77 есть последовательное выполнение стадий и этапов, определенных в таблице 2. В реальном жизненном цикле трудно провести четкую и определенную границу между этапами, а сам процесс создания ПО является итеративным: после завершения некоторого этапа почти всегда есть необходимость в коррекции уже выполненных этапов и стадий с целью внесения уточнений. При разработке принципиально нового ПО иногда бывает необходимо осуществить пробную реализацию с целью получения информации, требующейся для принятия решения на некоторой стадии.

Таблица 2.1

Стадии разработки

Этапы работ

Техническое задание

1. Обоснование необходимости разработки программ.
2. Выполнение научно-исследовательских работ (НИР).
3. Разработка и утверждение технического задания.

Эскизный проект

1. Разработка эскизного проекта.
2. Утверждение эскизного проекта.

Технический проект

1. Разработка технического проекта.
2. Утверждение технического проекта.

Рабочий проект

1. Разработка программы.
2. Разработка программной документации.
3. Испытание программы.

Внедрение

1. Подготовка и передача программы.

Специалистам в области разработки ПО известно, что наиболее важными стадиями в жизненном цикле разработки являются начальные, так как ошибки, допущенные на них, требуют значительных затрат на исправление на конечных стадиях.

Техническое задание. На стадии Техническое задание выполняются следующие работы, входящие в состав соответствующих этапов.

1. Обоснование необходимости разработки программ:

постановка задачи;

сбор исходных материалов;

выбор и обоснование критериев эффективности и качества;

обоснование необходимости проведения НИР.

2. Выполнение научно-исследовательских работ:

определение структуры входных и выходных данных;

предварительный выбор методов решения задач;
обоснование целесообразности применения ранее разработанных программ;
определение требований к техническим средствам;
обоснование принципиальной возможности решения поставленных задач.

3. Разработка и утверждение технического задания:

определение требований к программе;
разработка технико-экономического обоснования разработки программы;
определение стадий, этапов и сроков разработки программы и документации на нее;
выбор языков программирования;
определение необходимости проведения НИР на последующих стадиях;
согласование и утверждение ТЗ.

Результатом выполнения данной стадии является **техническое задание**, оформленное в соответствии с ГОСТ 19.105-78 (изм. 09.1981.) Общие требования к программным документам и ГОСТ 19.106-78 Общие требования к программным документам, выполненным печатным способом на листах формата 11 и 12 (по ГОСТ 2.301-68).

Эскизный проект. Основные этапы и содержание работ на стадии Эскизный проект приведены в таблице 2.2.

Таблица 2.2

Этапы работ

Содержание

Разработка ЭП

1. Предварительная разработка структуры входных и выходных данных.
2. Уточнение методов решения задач.
3. Разработка общего описания алгоритма решения задачи.
4. Разработка технико-экономического обоснования.

Утверждение ЭП

1. Разработка пояснительной записки.
2. Согласование и утверждение эскизного проекта.

Конкретное содержание работ стадии эскизного проекта и их объем определяет степень сложности разрабатываемого ПО. Результатом выполнения данной стадии является полное описание архитектуры ПО. Как правило, это описание делается на нескольких уровнях иерархии. На верхнем уровне детализации выделяются основные подсистемы, которым присваиваются имена, устанавливаются связи между подсистемами, их функции, получаемые путем декомпозиции предполагаемых функций ПО. Затем процедура декомпозиции выполняется для каждой подсистемы, выделяются модули, составляющие данную подсистему. В конечном итоге, получается иерархически организованная система, состоящая из уровней, каждый из которых представляет собой совокупность взаимосвязанных модулей.

Единицы, выделяемые на различных иерархических уровнях функциональной архитектуры системы, определяются по усмотрению разработчика. Стандарты ЕСПД различают программные единицы только с точки зрения их документирования.

Результаты эскизного проекта отображаются в документе Пояснительная записка к эскизному проекту, оформленному в соответствии с ГОСТ 19.105-78 и ГОСТ 19.404-79.

После утверждения пояснительной записи она становится программным документом, правила дублирования, учета, хранения которого определяется ГОСТ 19.601-78 Общие правила дублирования, обращения, учета и хранения и ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. Последующие стадии и этапы разработки ПО могут выявить необходимость внесения изменений в ЭП. Эти изменения должны быть отражены в пояснительной записке в соответствии с ГОСТ 19.603-78 Общие правила внесения изменений в программные документы и ГОСТ 19.602-78 Правила внесения изменений в программные документы, выполненные печатным способом.

В качестве примера, ниже приводится фрагмент расширенного описания работ стадии эскизного проекта.

1. Разработка эскизного проекта ПО.

разработка плана совместных работ на разработку ПО;

разработка и обоснование математической модели системы на ЭВМ и описание результатов моделирования;

разработка и обоснование алгоритмов и временных графиков функционирования ПО по всем режимам работы;

разработка и обоснование ресурсов памяти для реализации алгоритмов;

разработка перечня документов на ПО;

разработка и обоснование структуры БД, внешних входных и выходных данных;

разработка и обоснование алгоритмов информационного обеспечения;

определение взаимосвязей между видами программ;

разработка и обоснование набора тестов для проверки ПО;

разработка и обоснование организации наращивания и развития ПО;

оформление пояснительной записки и ведомости эскизного проекта ПО (в соответствии с ГОСТ 19.105-78, ГОСТ 19.404-79 и ГОСТ 2.106-68 ЕСКД. Текстовые документы);

согласование и утверждение ЭП.

Технический проект. Основные этапы и содержание работ на стадии Технический проект приведены в таблице 2.3.

Содержанием работ на этой стадии является проектирование структуры ПО. Результатом - реализующий заданный и утвержденный в техническом задании комплекс программ как иерархическая структура программных модулей, заданных своими функциональными спецификациями. Форма представления результата - Пояснительная записка к техническому проекту согласно ГОСТ 19.105-78, ГОСТ 19.404-79.

Разработка структуры ПО заключается в выделении всех программных компонентов по функциональным признакам, определение функциональных спецификаций модулей и уточнение внешних функциональных спецификаций, структуры входных и выходных данных, определении операционной среды, языковых средств и конфигурации аппаратных средств.

Спецификации модулей являются внешними характеристиками и содержат все сведения, необходимые вызывающим модулям. На последующих стадиях разработки спецификации оформляются в виде комментариев в начале текста исходной программы модуля. На данной стадии спецификации оформляются в виде комментария на принятом в организации, занимающейся разработкой ПО, языке спецификаций

Таблица 2.3.

Этапы работ

Содержание

Разработка ТП

1. Уточнение структуры входных и выходных данных.
2. Разработка алгоритмов решения задач.
3. Определение формы представления входных и выходных данных.
4. Определение синтаксиса и семантики языка.
5. Разработка структуры программы.
6. Окончательное определение конфигурации технических средств.

Утверждение ТП

1. Разработка плана мероприятий по разработке и внедрению программ.
2. Разработка пояснительной записки.
3. Согласование и утверждение ТП.

Фрагмент описания работ стадии технического проекта приведен ниже.

1. Разработка технического проекта ПО.

разработка плана совместных работ на разработку ПО;

распараллеливание задач применительно к возможностям ЭВМ и систем обмена информации и разработка организационной структуры программ;

формализация программно-аппаратных и внутризадачных интерфейсов;

разработка алгоритмов управления вычислительным процессом и процессами обмена, его исследование и оптимизация;

разработка и комплексное решение вопросов обеспечения устойчивой работы на программном уровне;

разработка модели и исследование на ней возможности реализации всех функций системы управления в рамках имеющихся вычислительных ресурсов на основе выбранного управляющего алгоритма и оптимизация структуры программ до получения удовлетворяющего результата;

уточнение структуры и определение формы представления БД, внешних входных и выходных данных;

разработка проекта описания программы для каждого вида программ ПО;

уточнение алгоритмов ПО и алгоритмов информационного обеспечения в процессе решения задач;

разработка проекта описания текстовых программ ПО;

разработка плана реализации ПО;

разработка проектов технических условий на ПО;

разработка и обоснование решений по надежности функционирования ПО;

разработка программы и правил испытаний ПО;

уточнение временных графиков функционирования программ ПО в процессе решения задач;

оформление пояснительной записки и ведомости технического проекта ПО (в соответствии с ГОСТ 19.105-78, ГОСТ 19.404-79 и ГОСТ 2.106-68 ЕСКД. Текстовые документы);

согласование и утверждение ТП.

Рабочий проект. Основные этапы и содержание работ на стадии Рабочий проект приведены в таблице 2.4.

Таблица 2.4

Этапы работ

Содержание

Разработка ПО

. Программирование и отладка программ.

Разработка программной документации

1. Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

Испытание ПО

1. Разработка, согласование и утверждение программ и методики испытаний
2. Проведение предварительных государственных, межведомственных приемо-сдаточных и других видов испытаний.
3. Корректировка ПО и программной документации по результатам испытаний.

Содержанием работ на этой стадии является описание ПО на выбранном проблемно-ориентированном языке (кодирование), отладка, разработка, согласование и утверждение порядка и методики испытаний, разработка программных документов, проведение тестирования, корректировка программ и программной документации по результатам тестирования, проведение приемо-сдаточных испытаний. Результат - ПО в форме программной документации, в форме документации на ПО или в форме программного изделия.

На стадии **Внедрения** осуществляется подготовка и передача ПО и программной документации для сопровождения и/или изготовления, оформление и утверждение акта о передаче ПО на сопровождение или изготовление, передача ПО в фонд алгоритмов и программ.

Кроме рассмотренного выше жизненного цикла программ, установленного 19 - ой системой стандартов, необходимо иметь в виду, что существует жизненный цикл автоматизированных систем, установленный

ГОСТ 34.601-90. Имеет смысл рассмотреть стадии и этапы разработки АС, так как 19-я система стандартов в настоящее время морально устарела и более современным представляется жизненный цикл АС. Анализ жизненных циклов разработки ПО, установленных ГОСТ 19.102 и ГОСТ 34.601, а также международных стандартов, регламентирующих данный процесс, будет приведен ниже.

Стандарт ГОСТ 34.601 распространяется на автоматизированные системы (АС), представляющие собой организационно-технические системы, обеспечивающую выработку решений на основе автоматизации информационных потоков в различных видах деятельности (исследование, проектирование, управление и т.п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях. Данный стандарт устанавливает стадии и этапы создания АС.

В процессе разработки АС создают, в общем случае, следующие виды обеспечения: техническое, программное, информационное математическое и др. Проектные решения по программному, техническому и информационному обеспечению реализуют как изделия в виде взаимоувязанной совокупности компонент и комплексов, входящей в состав АС с необходимой документацией. Справочное приложение к ГОСТ 34.601 устанавливает, что программное обеспечение АС - совокупность программ на носителях информации с программной документацией по ГОСТ 19.101. Таким образом, при разработке программного обеспечения можно пользоваться как стандартом ГОСТ 19.102, так и ГОСТ 34.601.

Процесс создания АС представляет собой совокупность упорядоченных во времени, взаимосвязанных, объединенных в стадии и этапы работ, выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям. Стадии и этапы создания АС выделяются как части процесса создания по соображениям рационального планирования и организации работ, заканчивающих заданным результатом.

Как и в стандарте ГОСТ 19.102, работы по развитию АС осуществляются по стадиям и этапам, применяемым для создания АС. Состав и правила выполнения работ на установленных настоящим стандартом стадиях и

этапах определяют в соответствующей документации организаций, участвующих в создании конкретных видов АС.

Стадии и этапы создания АС в общем случае приведены в таблице 2.5.

Стандартом ГОСТ 34.601 допускается исключать стадию Эскизный проект и отдельные этапы работ на всех стадиях, объединять стадии Технический проект и Рабочая документация в одну стадию Технорабочий проект. В зависимости от специфики создаваемых АС и условий их создания допускается выполнять отдельные этапы работ до завершения предшествующих стадий, параллельное во времени выполнения этапов работ, включение новых этапов работ.

Рассмотрим этапы, касающиеся разработки программного обеспечения.

При Обследование объекта и обоснование необходимости создания АС проводят: сбор данных об объекте автоматизации и осуществляемых видах деятельности; оценку качества функционирования объекта и осуществляемых видов деятельности, выявления проблем, решение которых возможно средствами автоматизации; оценку (технико-экономической, социальной и т.п.) целесообразности создания АС.

На этапе Формирование требований пользователя к АС проводят:

подготовку исходных данных для формирования требований к АС (характеристика объекта автоматизации, описание требований к системе, ограничения допустимых затрат на разработку, ввод в действие и эксплуатацию, эффект, ожидаемый от системы, условия создания и функционирования);

формулировку и оформление требований пользователя к АС.

Таблица 2.5.

Стадии

Этапы работ

1. Формирование требований к АС

1.1. Обследование объекта и обоснование необходимости создания АС

1.2. Формирование требований пользователя к АС

1.3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)

2. Разработка концепции АС

2.1. Изучение объекта

2.2. Проведение необходимых научно - исследовательских работ

2.3. Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя

2.4. Оформление отчета о выполненной работе

3. Техническое задание

- 3.1. Разработка и утверждение технического задания на создание АС
- 4. Эскизный проект
 - 4.1. Разработка предварительных проектных решений по системе и ее частям
 - 4.2. Разработка документации на АС и ее части
- 5. Технический проект
 - 5.1. Разработка проектных решений по системе и ее частям
 - 5.2. Разработка документации на АС и ее части
 - 5.3. Разработка и оформление документации на поставку изделий для комплектования АС и/или технических требований (технических заданий) на их разработку
 - 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации
- 6. Рабочая документация
 - 6.1. Разработка рабочей документации на систему и ее части
 - 6.2. Разработка или адаптация программ
- 7. Ввод в действие

- 7.1. Подготовка объекта автоматизации к вводу АС в действие
- 7.2. Подготовка персонала
- 7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)
- 7.4. Строительно-монтажные работы
- 7.5. Пусконаладочные работы
- 7.6. Проведение предварительных испытаний
- 7.7. Проведение опытной эксплуатации
- 7.8. Проведение приемочных испытаний
- 8. Сопровождение АС
 - 8.1. Выполнение работ в соответствии с гарантийными обязательствами
 - 8.2. Послегарантийное обслуживание

На этапе Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания) проводят оформление отчета о выполненных работах на данной стадии и оформление заявки на

разработку АС (тактико-технического задания) или другого заменяющего ее документа с аналогичным содержанием.

При Изучение объекта и Проведение необходимых НИР организация-разработчик проводит детальное изучение объекта автоматизации и необходимые НИР, связанные с поиском путей и оценкой возможности реализации требований пользователя, оформляют и утверждают отчеты о НИР.

Этап Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя направлен на разработку альтернативных вариантов концепции создаваемой АС и планов реализации; оценку необходимых ресурсов на их реализацию и обеспечение функционирования; оценку преимуществ и недостатков каждого варианта; сопоставление требований пользователя и характеристик предлагаемой системы и выбор оптимального варианта; определение порядка оценки качества и условий приемки системы; оценку эффектов, получаемых от системы.

На этапе Оформление отчета о выполненной работе подготавливают и оформляют отчет, содержащий описание выполненных работ на стадии, описание и обоснование предлагаемого варианта концепции системы.

На этапе Разработка и утверждение технического задания на создание АС проводят разработку, оформление, согласование и утверждение технического задания на АС и, при необходимости, технических заданий на части АС.

Результатом выполнения этапа Разработка предварительных проектных решений по системе и ее частям является: функции АС; функции подсистем, их цели и эффекты; состав комплекса задач и отдельных задач; концепции информационной базы, ее укрупненная структура; функции СУБД; состав вычислительной системы; функции и параметры основных программных средств.

На этапе Разработка проектных решений по системе и ее частям обеспечивают разработку общих решений: по системе и ее частям, функционально-алгоритмической структуре системы; по функциям персонала и организационной структуре; по структуре технических средств; по алгоритмам решения задач и применяемым языкам; по организации и ведению информационной базы, системе классификации и кодирования информации; по программному обеспечению.

На этапах 4.2 и 5.2 Разработка документации на АС и ее части проводят разработку, оформление, согласование и утверждение документации в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию АС.

На этапе Разработка и оформление документации на поставку изделий для комплектации АС и (или) технических требований (технических заданий) на их разработку проводят: подготовку и оформление документации на поставку изделий для комплектования АС; определение технических требований и составление ТЗ на разработку изделий, не изготавливаемых серийно.

На этапе Разработка заданий на проектирование в смежных частях проекта объекта автоматизации осуществляют разработку, оформление, согласование и утверждение заданий на проектирование в смежных частях проекта объекта автоматизации для проведения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием АС.

При выполнении этапа Разработка рабочей документации на систему и ее части осуществляют разработку рабочей документации, содержащей все необходимые и достаточные сведения для обеспечения выполнения работ по вводу АС в действие и ее эксплуатации, а также для поддерживания уровня эксплуатационных характеристик (качества) системы в соответствии с принятыми проектными решениями, ее оформление, согласование и утверждение. Виды документов - по ГОСТ 34.201.

На этапе Разработка и адаптация программ проводят разработку программ и программных средств

системы, выбор, адаптацию и (или) привязку приобретенных программных средств, разработку программной документации в соответствии с ГОСТ 19.101.

На этапе Подготовка объекта автоматизации к вводу АС в действие проводят работы по организационной подготовке объекта автоматизации к вводу АС в действие, в том числе: реализацию проектных решений по организационной структуре АС; обеспечение подразделений объекта управления инструктивно-методическими материалами; внедрение классификаторов информации.

Этап Подготовка персонала направлен на обучение персонала и проверку его способности обеспечить функционирование АС.

На этапе Комплектация АС поставляемыми изделиями обеспечивают получение комплектующих изделий серийного и единичного производства, материалов и монтажных изделий. Проводят входной контроль их качества.

На этапе Пусконаладочные работы проводят автономную наладку технических и программных средств, загрузку информации в БД и проверку системы ее ведения; комплексную наладку всех средств системы.

На этапе Проведение предварительных испытаний осуществляют: испытание АС на работоспособность и соответствие ТЗ в соответствии с программой и методикой предварительных испытаний; устранение неисправностей и внесение изменений в документацию на АС, в том числе эксплуатационную в соответствии с протоколом испытаний; оформление акта о приемке АС в опытную эксплуатацию.

На этапе Проведение опытной эксплуатации проводят: опытную эксплуатацию АС; анализ результатов опытной эксплуатации АС; доработку (при необходимости) программного обеспечения АС; дополнительную наладку (при необходимости) технических средств АС; оформление акта о завершении опытной эксплуатации.

На этапе Проведение приемочных испытаний проводят испытания на соответствие ТЗ в соответствии с программой и методикой приемочных испытаний, анализ результатов испытаний АС и устранение недостатков, выявленных при испытаниях, а также оформление акта о приемке АС в постоянную эксплуатацию.

На этапе Выполнение работ в соответствии с гарантийными обязательствами осуществляются работы по устранению недостатков, выявленных при эксплуатации АС в течение установленных гарантийных сроков, внесению необходимых изменений в документацию на АС.

На этапе Послегарантийное обслуживание осуществляются работы по анализу функционирования системы, выявлению отклонений фактических эксплуатационных характеристик АС от проектных значений, установлению причин этих отклонений, устранению выявленных недостатков и обеспечение стабильности эксплуатационных характеристик АС, внесению необходимых изменений в документацию на АС.

Требования к содержанию документов, разрабатываемых на каждой стадии и этапе, устанавливает руководящий документ РД-50-34.698-90. Кроме того, необходимо использовать для из разработки соответствующие стандарты Единой системы программной документации, Единой системы конструкторской документации и ГОСТ 34.602. Виды и комплектность документов регламентированы в ГОСТ 34.201.

3.1

Работа с CASE – средствами кодирования программного обеспечения

Цель работы: Подготовка модели к генерации программного кода; генерация программного кода

Теоретическая часть

Одно из самых мощных свойств Rational Rose – возможность генерации программного кода, представляющего модель. Варианты генерации программ меняются в зависимости от установленной версии Rose. В настоящее время применяются следующие три различные версии:

- Rose Modeler – позволяет создавать модель системы, но не поддерживает генерацию программного кода и обратное проектирование.
- Rose Professional – позволяет генерировать программный код на одном языке.
- Rose Enterprise – позволяет генерировать программный код на Ada 83, Ada 95, ANSI C++, CORBA, Java, COM, Visual Basic, Visual C++, C++ и XML. Кроме того, поддерживается генерация кода и обратное проектирование баз данных.

Лабораторный практикум построен на изучении Case-средства Rational Rose версии - Enterprise Edition.

Подготовка к генерации программного кода

Процесс генерации программного кода состоит из пяти основных этапов:

1. Проверка модели.
2. Создание компонентов.

3. Отображение классов на компоненты.
4. Установка свойств генерации программного кода.
5. Генерация программного кода.

В разных языках не все этапы обязательны. Так, программы, разработанные на языке C++, генерируются и без предварительного создания компонентов. Генерировать код программ на любом языке можно, не выполняя проверки модели, хотя во время генерации это порой приводит к различным ошибкам.

Хотя не все этапы обязательны, первые пять из них рекомендуется выполнять до начала процесса генерации. Проверка модели поможет выявить ее неточности и недостатки, нежелательные с точки зрения последующей генерации программ. Этапы, на которых рассматриваются компоненты, - это способ отобразить логическую схему системы на ее физическую реализацию, причем на этих этапах собирается большой объем полезной информации. Если пропустить эти этапы, то для создания компонентов Rose воспользуется пакетной структурой логического представления.

3.1 Проверка модели

В Rose существует не зависящее от языка средство проверки моделей, применяемое для обеспечения корректности модели перед генерацией программного кода. Такая проверка помогает выявить в модели неточности и ошибки, не позволяющие генерировать программы надлежащим образом.

В общем случае проверка модели может выполняться на любом этапе работы над проектом. Однако после завершения разработки графических диаграмм она является обязательной, поскольку позволяет выявить целый ряд ошибок разработчика. К числу таких ошибок и предупреждений относятся, например, не используемые ассоциации и классы, оставшиеся после удаления отдельных графических элементов с диаграмм, а также операции, не являющиеся именами сообщений на диаграммах взаимодействия.

Для проверки модели следует выполнить операцию главного меню: Tools Check Model (Инструменты проверить модель). Результаты проверки разработанной модели на наличие ошибок отображаются в окне журнала. Прежде чем приступить к генерации текста программного кода разработчику следует добиться устранения всех ошибок и предупреждений, о чем должно свидетельствовать чистое окно журнала.

К наиболее распространенным ошибкам относятся, например, сообщения на диаграмме Последовательности или Кооперативной диаграмме, не отраженные на операцию, либо объекты этих диаграмм, не отраженные на класс.

3.2 Создание компонентов

Второй этап процесса генерации программного кода - создание компонентов для классов. Существуют компоненты самых разных типов: файлы исходного программного кода, исполняемые файлы, библиотеки исполнения в реальном времени, компоненты ActiveX, апплеты и т.д. Перед генерацией программ можно отобразить каждый из классов на соответствующий компонент исходного программного кода.

После создания компонентов можно добавить зависимости между ними на диаграмме Компонентов. Зависимости между компонентами - это зависимости во время компиляции системы.

При генерации программ на C++, Java или Visual Basic выполнять подобный шаг не обязательно. В Java и Visual Basic Rose создаст автоматически соответствующий компонент для каждого из классов.

Для создания компонента:

1. Откройте диаграмму Компонентов (Component).
2. С помощью значка Component панели инструментов Diagram введите новый компонент в диаграмму.

3.3 Отображение классов на компоненты

Каждый компонент исходного кода - это файл с исходным программным кодом для одного или нескольких классов. В C++ каждый класс отображается на два компонента с исходным кодом: файл заголовка и основной файл (тело). В PowerBuilder на один компонент отображается несколько классов. Компонентом с исходным программным кодом в PowerBuilder является файл библиотеки PowerBuilder (.pbl). В Java каждый компонент - это один файл .java. Компоненты также создаются для элементов управления ActiveX, аплетов, файлов DDL, исполняемых файлов, а также других исходных и скомпилированных файлов.

Третий этап процесса генерации программного кода - отображение каждого из классов на соответствующие компоненты. В PowerBuilder необходимо отобразить каждый класс на компонент перед генерацией программы, в то время как в C++, Java и Visual Basic этот шаг не является обязательным. Rose может генерировать программный код самостоятельно. При генерации в Rose программ Java и Visual Basic производится еще и генерация нужных компонентов и отображение классов. Однако для C++ компоненты не создаются автоматически, а кроме того, ни для одного из языков не генерируются зависимости. Поэтому

рекомендуется выполнять этот шаг независимо от применяемого языка программирования. Для отображения класса на компонент:

- 1) Щелкните правой кнопкой мыши на компоненте, на диаграмме Компонентов или в браузере.
- 2) Выберите Open Specification в контекстном меню.
- 3) Выберите вкладку Realizes (Реализует).
- 4) Во вкладке Realizes щелкните правой кнопкой мыши на нужном классе (классах) и выберите Assign (Присвоить) в контекстном меню.
- 5) В браузере имя компонента будет показано в круглых скобках вслед за именем класса в Логическом представлении (Logical view).

ИЛИ

- 1) Найдите класс в окне Logical view браузера.
- 2) Перетащите класс на нужный компонент в окне Component view.
- 3) В окне Logical view имя компонента будет показано в круглых скобках за именем класса.

3.4 Установка свойств генерации программного кода

Для каждого языка в Rose предусмотрен ряд определенных свойств генерации программного кода. Можно установить несколько параметров генерации программного кода для классов, атрибутов, компонентов и других элементов модели. Этими свойствами определяется способ генерации программ. В Rose предлагаются общепринятые параметры по умолчанию.

3.4.1 Настройка свойств C++

Для того чтобы пользоваться возможностями C++, необходимо описать назначение этих свойств, список которых доступен во вкладке C++ спецификаций класса.

Перед генерацией программного кода рекомендуется анализировать эти свойства и вносить необходимые изменения.

3.4.2 Назначение свойств:

- 1) **CodeName** - устанавливает имя класса в создаваемом коде. Данное свойство необходимо устанавливать только в том случае, если имя класса должно быть отлично от имени заданного в модели Rational Rose. Данное свойство необходимо использовать для создания работоспособного кода C++, если для классов в модели используются русские имена.
- 2) **ImplementationType** - позволяет использовать простые типы вместо определения класса, устанавливаемого Rational Rose по умолчанию. При задании этого параметра создается директива `typedef`.
- 3) **ClassKey** - используется для задания типа класса, такого как `class`, `struct`, или `union`. Если тип не указан, то создается класс.
- 4) **GenerateEmptyRegion** - свойство указывает, как будет создаваться пустой раздел `protected: None` -

пустой раздел не будет создан; Preserved - пустой раздел будет создан, если будет установлено свойство «preserve=yes»; Unpreserved — пустой раздел будет создан, если будет установлено свойство «preserve=no»; All — всегда будет создаваться.

- 5) **PutBodiesInSpec** - если установлено как true, то в заголовочный файл попадет и описание тела класса. Используется для компиляторов, которым необходимо определение шаблона класса в каждом компилируемом файле.
- 6) **GenerateDefaultConstructor** - позволяет установить, необходимо ли создавать конструктор для класса по умолчанию. Может принимать следующие значения: DeclareAndDefine - создается определение для конструктора и скелет конструктора в теле класса; Declare Only - создается только определение; DoNotDeclare - не создается ни определения, ни скелета конструктора.
- 7) **DefaultConstructorVisibility** - устанавливает раздел, в котором будет определен конструктор по умолчанию: public, protected, private, implementation.
- 8) **InlineDefaultConstructor** - устанавливает, будет ли конструктор по умолчанию создаваться как inline подстановка. Если конструктора по умолчанию нет, то данное свойство не оказывает на код никакого эффекта.
- 9) **ExplicitDefaultConstructor** - устанавливает конструктор по умолчанию как explicit (явно заданный).
- 10) **InlineRelationalOperations** - определяет, будут ли функции операторов сравнения создаваться как inline подстановка.
- 11) **GenerateStorageMgmtOperations** - определяет, будут ли переопределяться операторы new и delete в классе.

- 12) **StorageMgmtVisibility** - определяет раздел, в который будут помещены операторы new и delete.
- 13) **InlineStorageMgmtOperations** - определяет, будут ли операторы new и delete определены как inline подстановка.
- 14) **GenerateSubscriptOperation** - определяет, будет ли переопределен оператор [].
- 15) **Subscript Visibility** определяет - раздел, в который будет помещен оператор [].
- 16) **SubscriptKind** - определяет вид функций оператора []: Common - обычная, Virtual - виртуальная, Abstract - абстрактная.
- 17) **SubscriptResultType** - определяет тип возвращаемого выражения для оператора [].
- 18) **InlineSubscriptOperation** - определяет, будет ли оператор [] определен как inline подстановка.
- 19) **GenerateDereferenceOperation** - определяет, будет ли переопределен оператор *.
- 20) **Dereference Visibility** - определяет раздел, в который будет помещен оператор *.
- 21) **DereferenceKind** - определяет вид функций оператора *: Common - обычная, Virtual - виртуальная, Abstract - абстрактная.
- 22) **DereferenceResultType** - определяет тип возвращаемого выражения для оператора *.

- 23) **InlineDereferenceOperation** - определяет, будет ли оператор * определен, как inline подстановка.
- 24) **GenerateIndirectionOperation** - определяет, будет ли переопределен оператор ->.
- 25) **IndirectionVisibility** - определяет раздел, в который будет помещен оператор ->.
- 26) **IndirectionKind** - определяет вид функций оператора ->: Common - обычная, Virtual - виртуальная, Abstract - абстрактная.
- 27) **IndirectionResultType** - определяет тип возвращаемого выражения для оператора ->.
- 28) **InlinelndirectionOperation** - определяет, будет ли оператор -> определен как inline подстановка.
- 29) **GenerateStreamOperations** - определяет, будут ли переопределены операторы потоков (« и »).

3.5 Выбор класса, компонента или пакета

При генерации программного кода за один раз можно создать класс, компонент или целый пакет. Программный код генерируется с помощью диаграммы или браузера. При генерации кода из пакета можно выбрать либо пакет Логического представления на диаграмме Классов, либо пакет представления Компонентов на диаграмме Компонентов. При выборе пакета Логического представления генерируются все его классы. При выборе пакета представления Компонентов генерируются все его компоненты.

Программный код можно генерировать одновременно для нескольких классов, компонентов или пакетов. На диаграмме с помощью клавиши Ctrl выберите классы, компоненты или пакеты, для которых нужно сгенерировать программный код, а затем - соответствующую команду генерации в меню.

3.6 Генерация программного кода

Если у вас установлены Rose Professional или Rose Enterprise, то в меню Tools предлагается несколько вариантов, специфичных для конкретного языка программирования.

Чтобы показать или скрыть эти пункты меню, выберите пункт Add-Ins □ Add-Ins (Надстройки □ Менеджер надстроек). В диалоговом окне Add-In Manager с помощью флажков покажите или скройте нужные варианты для различных языков.

Генерация программного кода в среде IBM Rational Rose 2003 возможна для отдельного класса или компонента. Для этого нужный элемент модели предварительно следует выделить в браузере проекта и выполнить операцию контекстного меню: Tools□C++□Code Generation - (Язык C++□Генерировать код). В результате этого будет открыто диалоговое окно с предложением выбора классов для генерации программного кода на выбранном языке программирования. После выбора соответствующих классов и нажатия кнопки OK программа IBM Rational Rose 2003 выполняет кодогенерацию.

Затем происходит компиляция и выдается окно статуса (Code Generation Status). Здесь можно увидеть информацию о том, какой класс был закодирован и количество ошибок и предупреждений. Если у вас произошла, какая-либо ошибка или же предупреждение, то их можно увидеть на рабочем поле в

Rational Rose, для этого и существует самое нижнее окно, в нем передаются все ваши действия и ошибки, произошедшие в ходе кодогенерации.

Рисунок 3.6 – Окно статуса компиляции

3.7 Результаты генерации

В результате кодогенерации Rational Rose создает два файла с расширением “.h” и “.cpp”, названия у них те же, что и название класса. Итак, выполнив эти действия, нажимаем правой клавишей на класс, появляется окошко, в нем ищем “C++”, и видим два пункта Browse Header и Browse Body, и в зависимости от того какой из файлов нам нужен “.h” (заголовочный) или “.cpp” (непосредственно реализация), выбираем их. Эти файлы открываются с помощью блокнота и теперь легко можно увидеть скелет класса, с различными комментариями, которые писали вы на диаграммах, и комментарии которые вставляет сама Rose. Теперь можно открыть один из файлов в C++ и доработать класс, описать работу функций, добавить различные нововведения.

Следует заметить, что при установленной на компьютер разработчика интегрированной среды сгенерированные файлы с текстом программного кода автоматически открываются в этой среде после двойного щелчка на пиктограмме этих файлов. Тем не менее, лучше копировать содержимое этих файлов в предварительно созданные программные проекты для полного контроля в этих средах процесса программирования и отладки приложений.

Сгенерированные программой IBM Rational Rose 2003 файлы с текстом программного кода содержат минимум информации. Для включения дополнительных элементов в программный код следует изменить свойства генерации программного кода, установленные по умолчанию.

В заключение следует отметить, что эффект от использования средства IBM Rational Rose 2003 проявляется при разработке *масштабных проектов* в составе команды или проектной группы. Однако ситуация покажется не столь тривиальной, когда станет необходимо выполнить проект с несколькими десятками вариантов использования и сотней классов. Именно для подобных проектов явно выявляется преимущество использования средства IBM Rational Rose 2003 и нотации языка UML для документирования и реализации соответствующих моделей.

Задание на лабораторное занятие

1. Изучить теоретический материал
2. Сгенерировать программный код на C++ для диаграммы классов, разработанной вами в предыдущей лабораторной работе.

Содержание отчета

- титульный лист;
- постановка задачи;
- листинг сгенерированного кода;
- ВЫВОД.

Контрольные вопросы

1. Какие диаграммы необходимо предварительно разработать, чтобы выполнить кодогенерацию?
2. Как посмотреть исходный код?
3. Какие установки свойств доступны на вкладке C++?
4. Какова структура создаваемого кода?
5. Что необходимо добавить в шаблоны классов для получения работоспособного приложения?
6. Какие шаги нужно предпринять для обновления модели по исходному коду?
7. Какие основные этапы кодогенерации вы знаете? Расскажите кратко о каждом из них?

3.2

Работа с CASE – средствами тестирования программного обеспечения

Цель работы: отладка разработанного программного средства; тестирование разработанного программного средства.

Теоретическая часть

Отладка ПС - это деятельность, направленная на обнаружение и исправление ошибок в ПС с использованием процессов выполнения его программ. *Тестирование ПС* - это процесс выполнения его программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор данных называется тестовым или просто тестом. Таким образом, отладку можно представить в виде многократного повторения трех процессов: тестирования, в результате которого может быть констатировано наличие в ПС ошибки, поиска места ошибки в программах и документации ПС и редактирования программ и документации с целью устранения обнаруженной ошибки.

Отладка программного средства невозможна без представления физической структуры. На этом этапе проектирования необходимо разработать *диаграмму компонентов*.

Диаграммой компонентов (Component diagram) называется диаграмма UML, на которой показаны компоненты системы и зависимости между ними.

Компонентом называется физический модуль кода. Компонентами бывают как библиотеки исходного кода, так и исполняемые файлы. Например, .h и .cpp и .exe - будут отдельными компонентами.

Особенности разработки диаграмм компонентов в среде IBM Rational Rose 2003

Диаграмма *компонентов* служит частью физического представления модели, играет важную роль в процессе ООАП и является необходимой для генерации программного кода. Для разработки диаграмм *компонентов* в браузере проекта предназначено отдельное представление *компонентов* (Component View), в котором уже содержится диаграмма *компонентов* с пустым содержанием и именем по умолчанию Main (Главная).

Активизация диаграммы *компонентов* может быть выполнена одним из следующих способов:

Щелкнуть на кнопке с изображением диаграммы *компонентов* на стандартной панели инструментов.

Раскрыть представление *компонентов* в браузере (Component View) и дважды щелкнуть на пиктограмме Main (Главная).

Через пункт меню Browse Component Diagram (Браузер Диаграмма *компонентов*).

В результате выполнения этих действий появляется новое окно с чистым рабочим листом диаграммы *компонентов* и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы *компонентов* (таблица 4.1).

Программа IBM Rational Rose 2003 не поддерживает следующие графические стереотипы. Графическое изображение этих стереотипов и их краткая характеристика приводятся в следующей таблице (таблица 4.2). При этом каждому из компонентов, как правило, соответствует отдельный файл исходной сборки

программного приложения.

Использование *стереотипов* существенно увеличивают наглядность графического представления диаграммы компонентов и позволяют архитектору уточнить характер реализации модели программистом на выбранном языке программирования.

Таблица 4.1 - Назначение кнопок специальной панели инструментов диаграммы компонентов

Selection Tool

Превращает изображение курсора в форму стрелки для последующего выделения элементов на диаграмме

Text Box

Добавляет на диаграмму текстовую область

Note

Добавляет на диаграмму примечание

Anchor Note to Item

Добавляет на диаграмму связь примечания с соответствующим графическим элементом диаграммы

Component

Добавляет на диаграмму компонент

Package

Добавляет на диаграмму пакет

Dependency

Добавляет на диаграмму отношение зависимости

Subprogram Specification

Добавляет на диаграмму спецификацию подпрограммы

Subprogram Body

Добавляет на диаграмму тело подпрограммы

Main Program

Добавляет на диаграмму главную программу

Package Specification

Добавляет на диаграмму спецификацию пакета

Package Body

Добавляет на диаграмму тело пакета

Task Specification

Добавляет на диаграмму спецификацию задачи

Task Body

Добавляет на диаграмму тело задачи

Generic Subprogram

Добавляет на диаграмму типовую подпрограммы(по умолчанию отсутствует)

Generic Package

Добавляет на диаграмму типовой пакет (по умолчанию отсутствует)

Database

Добавляет на диаграмму базу данных (по умолчанию отсутствует)

Таблица 4.2. Графическое изображение стереотипов компонентов и их характеристика

Subprogram Specification

Спецификация подпрограммы. Содержит описание переменных, процедур и функций и не содержит определений классов

Subprogram Body

Тело подпрограммы. Содержит реализацию процедур и функций, не относящихся к каким-то классам, при этом не содержит определений классов или реализаций операций других классов

Main Program

Главная программа. Реализует базовую логику работы программного приложения и содержит ссылки на другие компоненты модели

Package Specification

Спецификация пакета. Содержит определение класса, его атрибутов и операций. В языке программирования C++ спецификации пакета соответствует отдельный файл с расширением «h»

Package Body

Тело пакета. Содержит код реализации операций класса. В языке программирования C++ спецификации пакета соответствует отдельный файл с расширением «cpp»

Task Specification

Спецификация задачи. Может содержать определение класса, его атрибутов и операций, которые предполагается использовать в независимом потоке управления

Task Body

Тело задачи. Может содержать реализацию операций класса, которые имеют независимый поток управления.

Generic Subprogram

Типовая подпрограмма. Содержит описание переменных, процедур и функций, которые могут быть использованы в нескольких программных приложениях. При этом типовая подпрограмма не содержит определений классов

Generic Package

Типовой пакет. Содержит определение класса, его атрибутов и операций, которое может быть использовано в нескольких программных приложениях

Database

База данных. Содержит определение одного или нескольких классов, их атрибутов и, возможно, операций. При этом соответствующие классы могут быть реализованы в форме одной или нескольких таблиц базы данных

4.1 Добавление компонента на диаграмму компонентов и редактирование его свойств

Для добавления компонента на диаграмму компонентов нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы компонента на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. Добавить компонент на диаграмму можно также с помощью операции главного меню: Tools Create Component или с помощью операции контекстного меню: New Component, предварительно выделив представление компонентов в браузере проекта.

В результате этих действий на диаграмме появится изображение компонента с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию, которое разработчику следует изменить.

Для каждого компонента можно определить различные свойства, такие как стереотип, язык программирования, декларации, реализуемые классы. Редактирование этих свойств для произвольного компонента осуществляется с помощью диалогового окна спецификации свойств.

В частности, для компонента Main.exe можно выбрать стереотип <<EXE>> из предлагаемого вложенного списка, поскольку применительно к разрабатываемой модели предполагается реализация этого компонента в форме исполнимого файла. При этом на вкладке Realizes (Реализует) содержатся все классы, включая и актеров, которые на данный момент присутствуют в модели.

По умолчанию в среде IBM Rational Rose 2003 для всех добавляемых на диаграмму компонентов в качестве языка реализации используется язык анализа, который в последствии следует изменить на тот язык программирования, который предполагается использовать для написания программного кода. В

дальнейшем при генерации программного кода необходимо будет дополнительно выбрать те классы, которые реализует тот или иной компонент модели.

4.2 Добавление отношения зависимости и редактирование его свойств

Добавление отношения зависимости на диаграмму компонентов аналогично добавлению соответствующего отношения на диаграмму вариантов использования. Для добавления зависимости между двумя компонентами нужно с помощью левой кнопки мыши нажать кнопку с изображением зависимости на специальной панели инструментов, отпустить левую кнопку мыши, щелкнуть левой кнопкой мыши на изображении исходного компонента на диаграмме и отпустить ее на изображении целевого компонента. В результате этих действий на диаграмме появится изображение отношения зависимости в форме пунктирной линии со стрелкой, соединяющей два выбранных компонента.

5 Тестирование программного средства

Тестирование – процесс многократного повторения программы с целью обнаружения ошибок. Существуют следующие методы тестирования ПС:

- статическое тестирование (ручная проверка программы за столом);

- детерминированное тестирование (при различных комбинациях исходных данных);
- стохастическое (исходные данные выбираются произвольно, на выходе определяется качественное совпадение результатов или примерная оценка).

При тестировании разработанного ПС необходимо использовать подходящий по функциональности пакет прикладного математического программного обеспечения.

В выбранной среде необходимо произвести расчеты всех параметров, реализованных в ПС, затем следует сравнить результаты и сделать вывод о качестве данного программного продукта.

Задание на лабораторное занятие

1. Изучить теоретический материал
2. Выполнить тестирование и отладку информационной системы, разработанной в лабораторной работе, то есть разработать диаграмму компонентов рассматриваемого ПС.

Содержание отчета

- титульный лист;
- постановка задачи;

- | | |
|--|--|
| | <ul style="list-style-type: none"> - тестовый набор данных и результаты тестирования; - вывод. |
|--|--|

Перечень показателей текущего контроля

Номер показателя	Значение показателя
1.1.2.1	Способность проводить анализ процессов предметной
1.1.2.2	Способность создавать контекстные диаграммы
1.1.2.3	Способность создавать отчёты
1.1.2.4	Способность создавать диаграммы декомпозиции
1.1.2.5	Способность применять компоненты диаграммы декомпозиции
1.1.4.1	Способность использовать инструментарий BPwin при проектировании программного обеспечения
2.1.2.1	Понимание стадий и этапов разработки программного обеспечения
2.1.4.1	Способность применять инструментарий ERwin при проектировании программного обеспечения
2.1.4.2	Способность применять инструментарий ERwin при проектировании баз данных
2.1.4.3	Способность применять модели данных в моделях процессов BPwin
2.1.4.4	Уметь вычислять размер проектируемой базы данных
2.1.4.5	Способность применять прямое проектирование баз данных
2.1.7.1	Способность применять прямое проектирование баз данных

2.2.2.1	Способность применять прямое проектирование баз данных
3.1.2.1	Понимание стадий и этапов создания автоматизированных систем
3.1.4.1	Способность применять инструментарий Rational Rose при проектировании программного обеспечения
3.1.4.2	Способность генерировать программный код
3.1.4.3	Способность применять результаты кодогенерации
3.1.4.4	Понимание методов тестирования программного продукта
3.1.4.5	Способность применять методы тестирования и отладки программного обеспечения
3.1.7.1	Способность применять результаты кодогенерации
3.2.2.1	Способность применять методы тестирования и отладки программного обеспечения

2.3 Результаты освоения МДК.03.03 подлежащие проверке на текущем контроле

№ текущего контроля	Индекс занятия текущего контrollя	Проверяемая дидактическая единица	Формируемые компетенции	Основные показатели оценивания результата	№ задания относящийся к показателю оценивания	Метод контроля	Форма контроля	Вид контроля	Индексы занятий ранее изученных связанные с контролируемыми единицами
1	1.1.8	1.2	ПК.3.1, ПК.3.6	1.1.2.1, 1.1.2.2, 1.1.2.3, 1.1.2.4, 1.1.2.5,	1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7	Опрос	Практическая работа	Анализ выполнения задания	1.1.1, 1.1.7

				1.1.2.6, 1.1.2.7				
		1.11	ПК.3.1, ПК.3.6	1.1.11.1	1.3	Опрос	Практическая работа	Анализ выполнения задания
2	2.1.5	1.2	ПК.3.1, ПК.3.6	2.1.2.1	2.1	Опрос	Индивидуальн ые задания	Анализ выполнения задания
		1.5	ПК.3.1, ПК.3.6	2.1.5.1	2.2	Опрос	Индивидуальн ые задания	Анализ выполнения задания
		1.9	ПК.3.1, ПК.3.6	2.1.9.1	2.1	Опрос	Индивидуальн ые задания	Анализ выполнения задания
		1.10	ПК.3.1, ПК.3.6	2.1.10.1, 2.1.10.2, 2.1.10.3	2.1, 2.1, 2.1	Опрос	Практическая работа	Анализ выполнения задания
		1.11	ПК.3.1, ПК.3.6	2.1.11.1, 2.1.11.2, 2.1.11.3	2.2, 2.2, 2.2	Опрос	Практическая работа	Анализ выполнения задания
3	3.1.5	1.2	ПК.3.1, ПК.3.6, ПК.3.5	3.1.2.1	3.1	Опрос	Индивидуальн ые задания	Анализ выполнения задания
		1.5	ПК.3.1, ПК.3.6,	3.1.5.1	3.1	Опрос	Практическая работа	Анализ выполнения

		ПК.3.5					задания	
	1.9	ПК.3.1, ПК.3.6, ПК.3.5	3.1.9.1	3.2	Опрос	Практическая работа	Анализ выполнения задания	3.1.2, 3.1.4
	1.10	ПК.3.1, ПК.3.6, ПК.3.5	3.1.10.1, 3.1.10.2	3.3, 3.3	Опрос	Практическая работа	Анализ выполнения задания	3.1.1
	1.11	ПК.3.1, ПК.3.6, ПК.3.5	3.1.11.1, 3.1.11.2	3.4, 3.5	Опрос	Практическая работа	Анализ выполнения задания	

Перечень заданий текущего контроля

Номер задания	Задания
1.1	<p>3.1. Документы предварительных требований, спецификаций и ресурсов для разработки программного средства</p> <p>3.1.1. Интервью заказчиков и пользователей о проблемах и целях создания программного продукта:</p> <p>-определение профиля заказчика или пользователя:</p> <p>* имя, компания, отрасль;</p>

* что в основном производится;

* как измеряется успех деятельности пользователя;

* какие проблемы влияют на успешность деятельности пользователя;

-оценка проблемы создания или модификации программного средства;

* почему существует эта проблема;

* как она решается в настоящее время;

* как заказчик (пользователь) хотел бы ее решать;

-пользовательская среда:

* имеют ли пользователи опыт работы с данным типом программных средств;

* какая вычислительная платформа используется;

* каковы ожидания заказчика относительно практичности продукта;

* в каком виде должна быть представлена справочная информация для пользователя (в интерактивном или в виде печатной копии);

-предположения аналитика – разработчика относительно проблемы заказчика:

- * является ли проблема реальной;
- * каковы причины проблемы;
- * как она решается в настоящее время;
- * насколько важно для заказчика (пользователя) решение этой проблемы в сравнении с другими;
- оценка предлагаемого аналитиком метода решения проблемы;
- оценка возможности:
 - * кто в организации
 - заказчика нуждается в данном ПС;
- * сколько пользователей указанных типов может использовать ПС;
- оценка необходимого уровня надежности и производительности, а также потребности в сопровождении программного продукта:
- * каковы ожидания относительно надежности;
- * какой должна быть производительность;
- * каковы требования безопасности применения ПС;

- * *какие требования относительно установки и конфигурации;*
- * *существуют ли специальные требования по лицензированию;*
- * *существуют ли законодательные требования, требования внешней среды, инструкции или другие регламентирующие стандарты, которых необходимо придерживаться;*
- заключение аналитика – потребности или проблемы, с наивысшими приоритетами, выявленные в беседе с заказчиком (пользователем).

3.1.2. Результаты обследования и описание системы и целей разработки комплекса программ:

-характеристики существующей системы информатизации или управления:

- * *результаты её функционирования;*
- * *тенденции развития;*
- * *требования к номенклатуре и качеству результатов функционирования;*
- * *характеристики взаимодействия системы с внешней средой;*
- * *существующие характеристики качества системы и тенденции их изменения во времени;*

-описание существующей системы:

- * *функциональной и информационной структуры системы;*

* качественных и количественных характеристик, раскрывающих взаимодействие ее компонентов в процессе функционирования;

- описание недостатков существующей системы:

* результаты диагностического анализа, при котором оценивается качество функционирования и организационно-технологический уровень системы;

* недостатки в организации и технологии функционирования информационных процессов системы;

* степень их влияния на качество функционирования системы;

- обоснование необходимости совершенствования системы:

* степень соответствия показателей функционирования системы предъявляемым современным требованиям;

* степень соответствия прогнозируемых характеристик требуемым;

* необходимость совершенствования системы путем создания нового или модернизации существующего ПС;

- цели, критерии и ограничения создания нового ПС:

* производственно-хозяйственные, научно-технические и экономические цели создания ПС;

* критерии оценки эффективности создания ПС;

* ограничения ресурсов при создании нового ПС;

-функции и задачи создаваемого, нового ПС:

* обоснование выбора перечня автоматизируемых функций и комплексов задач;

* возможная очередность внедрения функциональных задач;

* требования к характеристикам реализации функций и задач;

-ожидаемые технико-экономические результаты создания ПС:

* перечень основных источников экономической эффективности, получаемых в результате создания ПС;

* оценка ожидаемых изменений основных технико-экономических и социальных показателей функционирования системы;

* оценка ожидаемых затрат на создание и эксплуатацию ПС с распределением их по очередям создания системы и по годам;

* ожидаемые обобщающие показатели экономической эффективности системы с новым ПС;

-выводы и предложения:

* о производственно-хозяйственной необходимости и технико-экономической целесообразности создания нового ПС;

* о совершенствовании организации и технологии процесса деятельности системы и ПС;

* обобщенные рекомендации по созданию нового или модернизированного ПС.

3.1.3. Технико-экономическое обоснование проекта программного средства:

-базовые стандарты, методы, правила и инструментальные средства, которые могут быть использованы при разработке требований к программному продукту;

-стандарты и методы, которые должны быть применены при разработке требований к структуре и компонентам ПС;

-системы основных обозначений, которые следует применять для описания требований, диаграмм потока данных и формальные языки спецификаций;

-общие исходные данные:

* класс и общие функции проекта ПС;

* цели анализа и возможная достоверность исходных данных;

* новизна проекта и функций комплекса программ;

* необходимая степень согласованности проекта с требованиями технического задания заказчика;

* возможность управления рисками и архитектурой проекта ПС;

* требуемый уровень обобщенной слаженности и организованности коллективной разработки проекта;

* возможный уровень обеспечения и оснащения технологии разработки по методике СММ;

* наличие значительного ограничения сроков разработки ПС;

- выбор методики и сценария первичной оценки требований технико-экономических характеристик ПС:

* оценка размера - масштаба программного продукта проекта;

* оценка доли возможного использования готовых программных компонентов;

- предварительный расчет трудоемкости и стоимости разработки ПС;

- предварительный расчет длительности разработки ПС;

- предварительный расчет необходимого числа специалистов;

- оценка влияния технико-экономических факторов на качество проекта ПС:

* оценка влияния требования надежности ПС;

* оценка влияния требований высокой степени соответствия документации программному продукту;

* оценка влияния уровня и стабильности коллектива специалистов проекта ПС;

* тематическая квалификация специалистов;

* технологическая квалификация проектировщиков и программистов;

- * оценка влияния технологической среды разработки на технико-экономические показатели проекта ПС;
 - * оценка влияния вычислительной среды и ограничения ресурсов объектной ЭВМ реального времени на технико-экономические показатели проекта ПС;
 - * оценка влияния стабильности требований заказчика к задачам и функциям ПС;
- предварительная оценка результирующих технико-экономических характеристик проекта ПС:
- * расчет уточненной трудоемкости и стоимости разработки проекта программного продукта;
 - * расчет уточненной длительности разработки проекта ПС;
 - * уточненный расчет необходимого числа специалистов для разработки проекта ПС;
 - * расчет уточненного распределения трудоемкости, длительности, числа специалистов по этапам работ;
- обобщение основных технико-экономических характеристик и полной стоимости разработки проекта ПС;
- анализ результатов и технико-экономическое обоснование целесообразности продолжения проектирования программного продукта.

3.1.4. Концепция и основные предложения по созданию базовой версии программного средства:

- описание обобщенных результатов обследования и изучения существующей системы и внешней среды;
- перечень базовых стандартов проекта программного продукта;

- описание потребностей потенциальных пользователей ПС;
- характеристики комплекса задач:
 - * *назначение комплекса задач;*
 - * *перечень объектов (технологических объектов управления, подразделений предприятия и т. п.), при управлении которыми решают комплекс задач;*
 - * *периодичность и продолжительность решения;*
 - * *связи данного комплекса задач с внешней средой и другими комплексами системы;*
 - * *распределение функций между персоналом, программными и техническими средствами при различных ситуациях решения комплекса задач;*
- входная информация:
 - * *источники информации и их идентификаторы;*
 - * *перечень и описание входных сообщений (идентификаторы, формы представления, сроки и частота поступления);*
 - * *перечень и описание структурных единиц информации входных сообщений или ссылка на документы, содержащие эти данные;*
- выходная информация:

* получатели и назначение выходной информации;

* перечень и описание выходных сообщений;

* периодичность их выдачи;

* допустимое время задержки решения.

- описание и оценка преимуществ и недостатков разработанных альтернативных вариантов функций в концепции создания ПС;

-сопоставительный анализ требований пользователя к ПС и вариантов функций в концепции ПС по удовлетворению требований заказчика и пользователей;

-обоснование выбора оптимального варианта содержания и приоритетов комплекса функций в концепции;

-общее описание постановки выбранных задач и функций нового ПС;

-предполагаемая структура, состав компонентов и интерфейсы с внешней средой;

-ожидаемые результаты и возможная эффективность реализации выбранного варианта концепции ПС;

-ориентировочный план реализации выбранного варианта концепции ПС;

-требования к составу и содержанию документации проекта ПС;

-оценка необходимых затрат ресурсов на разработку, ввод в действие и обеспечение функционирования

нового ПС;

-предварительный состав требований, гарантирующих качество ПС;

-предварительные условия испытаний и приемки системы.

3.1.5.Предварительный укрупненный план проектирования и разработки базовой версии программного средства:

-общее назначение и сфера действия плана;

-перечень и план-график взаимосвязи этапов и работ;

-для каждой работы, выделенной в соответствии со стандартом жизненного цикла ПС, и включенной в план:

* *наименование работы;*

* *контролируемые характеристики объекта разработки;*

* *контролируемые показатели процесса выполнения плана;*

* *требования к результатам и качеству;*

* *состав отчетной документации о результатах выполнения плана и достигнутых показателях качества;*

* *дата начала и окончания работы;*

* *наименование специалистов и подразделений - участников работы;*

* *фамилия и должность ответственного исполнителя этапа работ;*

-оценка возможности, этапы и сроки реализации конкретных требований концепции и решения функциональных задач.

3.1.6. Системный проект, общее описание программного средства и среды разработки для согласования между заказчиком и разработчиком:

-основные ведения о техническом, информационном и других видах внешней среды, необходимые для разработки программного средства;

-ссылки на документы проекта системы, влияющие на разработку конкретного программного средства;

-структура программного средства – перечень функциональных частей и компонентов с указанием их взаимосвязей и обоснованием выделения каждой из них;

-функции частей программного средства – назначение и описание основных проблем и функций для каждой части и компонента ПС;

-содержание решаемой проблемы:

* *соглашение с заказчиком по определению проблемы;*

* *заинтересованные лица и пользователи;*

* *границы системы решений;*

* *ограничения, которые необходимо наложить на решения;*

-содержание потребностей заказчика и пользователей:

* *структурированное интервью, используя 10 – 15 наиболее часто упоминавшихся потребностей и функций пользователей;*

* *сформулированные описания потребностей для начала создания комплекса требований;*

* *классификация и приоритеты функций, чтобы определить их как критические, важные и полезные;*

-категории критичности системы и программного средства:

* *критическое ПС – для него проявление дефекта при испытаниях или возникновение отказовой ситуации прекращает безопасное функционирование системы обработки информации и резко увеличивает риск для жизни и здоровья людей или риск аварии оборудования с большим ущербом;*

* *важное ПС – для которого проявление дефекта или возникновение отказовой ситуации может существенно снизить качество выпускаемой продукции и заметно увеличить риск и цену грубых ошибок при обработке информации;*

* *ординарное ПС – для которого ошибки или отказовые ситуации отражаются только на качестве и надежности выходных результатов и не могут привести к значительному ущербу в объектах внешней среды или к потерям у пользователей.*

-определение системы и программного продукта:

- * определение программного продукта, согласование с участниками проекта и их подтверждение;
- * атрибуты приоритетов функций (критический, важный, полезный), риска (высокий, низкий, средний);
- * иллюстративные прецеденты в приложении к концепции, чтобы его функции были понятны заказчикам и пользователям;

-управление и согласование масштаба проекта и контроль изменений:

- * соглашение с заказчиком относительно масштаба проекта и решений по изменению масштаба;
- * фиксация новых функций, возникающих в результате нормального течения проекта, контроль за изменениями и решениями;
- * система управления запросами изменений, чтобы гарантировать, что они поступают через контроль за изменениями.

-построение корректной системы и комплекса программ:

- * анализ и оценка рисков, чтобы определить, план действий по верификации и проверке корректности требований, основываясь на результатах этих оценок;
- * тестовые процедуры и примеры, которые трассируются к прецедентам, а также к функциональным и конструктивным требованиям;

- * *периодическое проведение приемочных испытаний компонентов, чтобы проверять правильность частных результатов работы и обеспечить постоянное участие заказчиков;*
- непрерывное управление процессом работы с требованиями:
- * *лидер – менеджер должен нести персональную ответственность за концепцию, оценивать состояние дел и регулярно готовить отчеты и запросы, отражающие эти действия;*
- * *формализация и отслеживание процесса изменения требований к программному средству, чтобы убедиться, что концепция надлежащим образом осуществляется в детализированных требованиях технического задания.*

3.1.7. Техническое задание на предварительное (детальное) проектирование программного средства:

- общие сведения:
 - * *титульный лист с утверждающими и согласующими подписями;*
 - * *полное наименование проекта ПС;*
 - * *назначение и цель разработки (развития) ПС;*
 - * *основание для выполнения и финансирования проекта ПС;*
 - * *организация - заказчик проекта;*
 - * *организация - головной исполнитель и соисполнители проекта;*

* общие сроки выполнения проекта;

-общие технические требования, стандарты и базовые нормативные документы для выполнения проекта ПС;

-характеристики системы информатизации или управления;

* общие требования к системе и внешней среде;

* общие требования от системы к структуре программного средства:

* требования к программному средству в целом;

* требования к функциям и основным задачам проекта ПС;

* требования к внешней среде проекта ПС;

* требования к классам и характеристикам пользователей;

-детальные спецификации требований к функциям, компонентам и эксплуатационным характеристикам ПС:

* требования к структуре и функционированию ПС;

* требования к надежности и безопасности применения;

* требования к защите информации;

* требования к стандартизации и унификации;

- * требования к интерфейсам между компонентами, с внешней средой и с пользователями;
- специальные требования к аппаратной и операционной платформам для реализации ПС;
 - требования к содержанию, оформлению и обозначениям эксплуатационной и технологической документации;
 - требования к составу и содержанию работ по внедрению ПС в эксплуатацию;
 - этапы, сроки и график выполнения основных работ;
 - ожидаемые результаты применения ПС и форма их представления;
 - порядок контроля, испытаний и приемки результатов проекта;
 - предложения по применению и развитию проекта ПС.

1.2

3.2. Документы процессов проектирования и выбора характеристик качества программного средства

3.2.1. Стандарты и ограничения на процессы проектирования программного средства:

- стандарты, методы, правила и описания инструментальных средств, которые следует применять при

- разработке архитектуры ПС и требований компонентов нижнего уровня;
- стандарты и общие методы описания проекта ПС, которые будут использованы;
- соглашения по идентификации компонентов, версий и продуктов проекта программного средства;
- ограничения, налагаемые на применяемые методы проектирования;
 - * *ограничения при распределении ресурсов трудоемкости, стоимости, времени, специалистов;*
 - * *ограничения использования прерываний и структур, управляемых событиями;*
 - * *возможности использование динамических задач;*
 - * *пределы использования глобальных данных;*
- ограничения на использования инструментальных средств автоматизации проектирования;
- ограничения на проектирование структуры программ – запрещение использования рекурсий, динамических объектов, альтернативных имен, сокращенных выражений;
- ограничения по сложности – максимальный уровень вложенности циклов, последовательных вызовов и условных структур, использования безусловных переходов, число входных/выходных точек компонентов и модулей программы.

3.2.2. Спецификация требований к системе и к комплексу программ:

- требования проекта системы комплекса программ как целого и общей архитектуры системы;
- требования к унификации проекта интерфейсов и базы данных;
- требования и обоснование выбора проектных решений уровня системы, выбора компонентов системы, описание поведения системы с точки зрения пользователя;
- спецификация требований верхнего уровня, производные требования к компонентам ПС и требования к интерфейсам между системными компонентами, элементами конфигурации ПС и аппаратуры;
- описание распределения системных требований по компонентам ПС с учетом требований, которые обеспечивают характеристики качества;
- требования к архитектуре системы, содержащей идентификацию и функции компонентов системы, их назначение, статус разработки, аппаратные и программные ресурсы;
- требования концепции совместного целостного функционирования компонентов ПС, описание их динамических связей;
- требования стабильности интерфейсов между аппаратными и программными компонентами и пользователями;
- требования возможности анализа трассируемости компонентов программного средства системы к системным требованиям проекта;
- требования для системы или/и подсистем и методы, которые должны быть использованы для гарантии того, что каждое требование будет выполнено и прослеживаемо к конкретным требованиям системы:

- к режимам работы;
- к производительности системы;
- к внешнему и пользовательскому интерфейсу системы;
- к внутреннему интерфейсу системы;
- к внутренним данным системы;
- по возможности адаптации к внешней среде;
- по обеспечению безопасности системы и внешней среды;
- по обеспечению защиты и секретности данных;
- к доступным ресурсам вычислителя, к коэффициенту использования ресурсов аппаратуры, к организации сети компьютеров, если это необходимо;
- по ограничениям доступных ресурсов проекта;
- по обучению и уровню квалификации персонала;
- по возможностям средств аттестации результатов и компонентов, включающие в себя демонстрацию, тестирование, анализ, инспекцию и требуемые специальные методы для контроля функций и качества конкретной системы или компонента.

3.2.3. Предварительное описание и контроль согласованности требований компонентов проекта программного средства:

- описание архитектуры и контроль согласованности требований компонентов нижнего уровня ПС, которые должны удовлетворять требованиям верхнего уровня к комплексу программ;
- детализированное описание того, как компоненты ПС удовлетворяет специфицированным требованиям верхнего уровня к ПС, включая алгоритмы, структуры данных, и описание распределения требований по процессорам и задачам для реализации;
- описание архитектуры комплекса программ в составе системы, которая определяет структуру компонентов ПС, предназначенного для реализации заданных требований на систему;
- описание входных/выходных данных для внутренних и внешних интерфейсов архитектуры системы и комплекса программ;
- анализ корректности описаний потоков данных и потоков управления;
- ограничения на использование ресурсов, стратегия для управления каждым ресурсом, границы рабочего диапазона и методы измерения времени выполнения и использования памяти при функционировании ПС;
- обоснование тех решений проекта, которые относятся к перечисленным требованиям, связанным с применением системы и ПС.

3.2.4. Описание функционирования программного средства, взаимодействия с объектами внешней среды и человека-машинного диалога:

-исходные данные:

* перечень исходных материалов и документов, использованных при разработке функциональной части проекта ПС;

* особенности системы, влияющие на проектные решения и компоненты ПС по автоматизированным функциям;

* данные о системах, взаимосвязанных с разрабатываемым ПС;

* сведения об информации, которой ПС должно обмениваться с абонентами и другими системами;

* описание информационной модели системы;

-цели комплекса программ и автоматизируемые функции;

-характеристика функциональной структуры:

* перечень компонентов ПС с указанием функций и задач, реализуемых в каждом компоненте;

* описание процессов выполнения функций;

* необходимые пояснения и обоснования к разделению автоматизированных функций на операции, выполняемые техническими, программными средствами и человеком;

* требования к временному регламенту и характеристикам процесса реализации автоматизированных функций (точности, надежности и т.п.) при решении требуемых задач;

-типовые решения с указанием функций и задач, для выполнения которых они применены;

-функциональная пригодность:

* *соответствие структурных характеристик комплекса программ назначению и функциям ПС;*

* *соответствие назначения целям применения ПС;*

* *соответствие требований к заданным функциям назначению ПС;*

* *соответствие исходной информации требованиям к функциям ПС;*

* *соответствие состава и содержания выходной информации для потребителей назначению и функциям ПС.*

3.2.5. Описания алгоритмов компонентов (модулей) программного средства:

-назначение и характеристики компонента (модуля) ПС:

* *постановка задачи, для решения которой компонент предназначен;*

* *краткие сведения о процессе (объекте), при управлении которым используется алгоритм;*

* *воздействия на процесс пользователями, осуществляемые при функционировании алгоритма;*

* *ограничения на возможность и условия применения алгоритма;*

* *требуемые характеристики качества решения;*

* состав и общие требования к входным и выходным данным;

-используемая информация и ее характеристики:

* массивы информации, сформированные из входных сообщений системы;

* массивы информации, полученные в результате работы других алгоритмов и сохраняемые при реализации данного алгоритма;

-результаты решения:

* перечень и содержание массивов информации, формируемых в результате реализации алгоритма;

-математическое описание:

* математическая модель или экономико-математическое описание процесса (объекта);

* перечень принятых допущений и оценки степени соответствия принятой модели реальному процессу (объекту);

* сведения о результатах научно-исследовательских работ, если они использованы для разработки алгоритма;

* описание логики функционирования алгоритма и способа формирования результатов решения;

* последовательности этапов счета, расчетные и логические формулы, используемые в алгоритме;

-требования к разработке программы для реализации алгоритма:

- * диагностические сообщения при работе с программой;
- * требования к контролю данных в процессе выполнения операций алгоритма;
- * ограничения, связанные с машинной реализацией программы алгоритма;
- * требования к контрольной задаче алгоритма.

3.2.6. Описание информационного обеспечения программного средства и системы управления базами данных:

- состав информационного обеспечения комплекса программ;
- наименования и назначения всех баз и наборов данных;
- проектные решения, связанные с применением базы данных с точки зрения пользователя;
- интерфейсы базы данных с другими системами, элементами конфигурации ПС и аппаратуры;
- способы доступа к базам данных, время реакция баз данных на входные запросы и другие эксплуатационные характеристики;
- алгоритмы работы с базами данных и возможные ограничения;
- организация интерфейсов и информационного обеспечения программного средства:
 - * принципы организации информационного обеспечения системы и ПС;

- * обоснование выбора носителей данных и принципы распределения информации по типам носителей;
- * описание видов и методов контроля данных при создании и функционировании информационных баз;
- * описание решений и интерфейсов, обеспечивающих информационную совместимость ПС с другими системами по источникам и потребителям информации;

-организация сбора и передачи информации:

- * перечень источников и носителей информации;
- * оценки интенсивности и объема потоков информации;
- * требования к интерфейсам и организации сбора, передачи, контроля и корректировки информации;

-система классификации и кодирования информации в комплексе программ:

- * описание классификации объектов в действующих классификаторах;
- * методы кодирования объектов во вновь разработанных классификаторах;

-организация информации баз данных:

- * характеристики состава и объема баз данных;
- * описание структуры баз данных;

- * описание взаимосвязей баз данных и функций ПС, при реализации которых используется каждая база данных;
- организация ведения и изменения каждой информационной базы:
- * последовательность процедур при создании и обслуживании базы данных;
- * регламент выполнения процедур и средств защиты базы данных от разрушения и несанкционированного доступа;
- * связи между массивами баз данных и массивами входной информации системы и ПС.

3.2.7. Требования к характеристикам качества проекта программного средства:

-требования к корректности программного средства:

- * соответствие требований к функциям ПС требованиям к системе;
- * соответствие требований к функциональным компонентам ПС требованиям к функциям комплекса программ;
- * соответствие текстов программ требованиям к функциональным компонентам ПС;
- * соответствие объектного кода исходному тексту программ функциональных компонентов ПС;
- * степень покрытия тестами возможных маршрутов исполнения программ;

-требования к способности и качеству взаимодействия компонентов системы и ПС:

* с операционной системой;

* с аппаратной средой;

* с внешней средой системы и с пользователями;

* между программными компонентами;

* между компонентами распределенных информационных систем;

-требования к защищенности – безопасности ПС:

* соответствие стандартам и нормативным документам на защиту от различных типов угроз;

* соответствие критериям и требованиям защиты от предумышленных несанкционированных угроз безопасности ПС;

* соответствие методам и средствам защиты от проявления случайных дефектов программ и данных;

* обеспечение эффективности оперативных методов защиты и восстановления при проявлениях и реализации угроз безопасности;

* обеспечение равнопрочной защиты в соответствии с опасностью угроз и доступностью ресурсов для защиты;

-требования к характеристикам надежности функционирования ПС:

- * к завершенности – наработке на отказ при отсутствии рестарта;
 - * к устойчивости – наработке на отказ при наличии автоматического рестарта на обеспечение надежности;
 - * к восстанавливаемости – допустимой длительности восстановления;
 - * к доступности-готовности – относительному времени работоспособного функционирования;
- требования к эффективности функционирования ПС:
- * к временной эффективности: допустимое время отклика – получения результатов на типовое задание;
 - * к пропускной способности – числу типовых заданий, исполняемых в единицу времени;
 - * к используемости ресурсов: относительная величина использования ресурсов ЭВМ при нормальном функционировании программного средства;
- требования к практичности применения ПС:
- * к понятности: четкость концепции ПС; демонстрационные возможности; наглядность и полнота документации;
 - * к простоте использования: простота управления функциями; комфортность эксплуатации; среднее время ввода заданий; среднее время отклика на задание;
 - * к изучаемости: трудоемкость изучения применения ПС; продолжительность изучения; объем эксплуатационной документации; объем электронных учебников;

-требования к сопровождаемости ПС:

* к анализируемости: стройность архитектуры программ; унифицированность интерфейсов; полнота и корректность документации;

* к изменяемости: трудоемкость подготовки изменений; длительность подготовки изменений;

* к тестируемости: трудоемкость тестирования изменений; длительность тестирования изменений;

-требования к мобильности ПС :

* к адаптируемости: трудоемкость адаптации; длительность адаптации;

* к простоте установки: трудоемкость инсталляции; длительность инсталляции.

* к замещаемости: трудоемкость замены компонентов; длительность замены компонентов.

3.2.8. Пояснительная записка к предварительному или детальному проекту программного средства:

-общие положения:

* наименование проектируемой системы и наименования документов, их номера и дату утверждения, на основании которых ведется проектирование ПС;

* цели, назначение и области использования ПС;

* перечень организаций, участвующих в разработке системы и ПС;

- * сроки выполнения основных этапов и проекта ПС в целом;
- * сведения об использованных при проектировании нормативных документах;
- * сведения о НИР и изобретениях, использованных при разработке проекта;
- * очередность создания функциональных компонентов комплекса программ и объем каждой очереди;
- описание процесса функционирования и применения ПС:
- * состав функций и операций применения ПС с учетом обеспечения взаимосвязи и совместимости процессов деятельности пользователей;
- * требования к организации работ пользователей в условиях функционирования системы;
- основные технические решения:
- * структура ПС в целом и компонентов, средства и способы взаимодействия для информационного обмена между компонентами;
- * условия связи ПС с взаимодействующими системами, обеспечение их совместимости;
- * функции комплексов задач, реализуемых ПС и компонентами;
- * состав обрабатываемой информации, размер и способы ее организации;
- * режимы функционирования, диагностирования работы ПС и компонентов;

- * виды машинных носителей, входные и выходные документы и сообщения, последовательность обработки информации;
- * состав инструментальных программных средств, технология, языки проектирования и их реализация;
- * численность, квалификация и функции обслуживающего персонала и возможных пользователей ПС, режимы работы, порядок взаимодействия специалистов;
- * сведения о реализации заданных в спецификациях требований к потребительским характеристикам ПС и компонентов, определяющих их качество и безопасность применения;
- описание организации баз данных компонентов и процессов проектирования версий программного средства;
- описание логической и физической структуры баз данных проектирования компонентов и ПС в целом:
- * состав, форматы и содержание данных версий компонентов и процессов проекта ПС;
- * взаимосвязи между компонентами и версиями проекта ПС и данными;
- * описание принятого варианта расположения компонентов проекта ПС и данных на конкретных машинных носителях;
- мероприятия по подготовке системы и ПС к вводу в действие и к эксплуатации:
- * мероприятия по адаптации и приведению внешней, исходной информации к виду, пригодному для обработки ПС;

* мероприятия по обучению и проверке квалификации обслуживающего персонала и пользователей;

* мероприятия по организации необходимых подразделений и рабочих мест пользователей.

3.2.9. Описание концепции технологии автоматизированного проектирования программного средства:

-общие положения:

* класс процессов и этапов жизненного цикла ПС, на которые распространена технология;

* состав и квалификация специалистов-пользователей технологии;

* требования и ограничения на условия применения технологии;

-постановка задач и цели автоматизации проектирования ПС:

* основные пути и направления решения задач проектирования ПС;

* требования к качеству технологии проектирования ПС;

* ограничения на применение технологии проектирования;

* критерии оценки качества результатов применения технологии;

-методика проектирования ПС:

* математические методы, используемые при проектировании ПС;

* состав и назначение проектных процедур;

* порядок взаимодействия проектных процедур в процессе их выполнения в жизненном цикле ПС;

-инструментальные средства:

* состав и функциональные возможности выбранных средств автоматизации проектирования и обеспечения жизненного цикла ПС;

* рекомендуемые функции инструментальных средств;

* адаптация и инсталляция инструментальных средств автоматизации проектирования к конкретным условиям проекта;

* ссылки на инструкции и их разделы по использованию инструментальных средств проектирования;

-исходные данные для использования технологии:

* состав, порядок выбора, представления и формирования массивов используемой исходной информации проекта;

* перечень и обозначения компонентов, описывающих предметную область, с указанием их наименований, единиц измерений, диапазонов изменения значений;

* критерии оценки качества исходных данных проекта ПС;

-описания проектных процедур в жизненном цикле ПС:

* *состав исходных данных;*

* *правила доступа к ним;*

* *порядок выполнения проектных процедур;*

* *состав и форма результирующих сообщений;*

-оценка качества результатов проектирования:

* *анализ полученных проектных решений на соответствие заданным требованиям спецификаций и критериям качества.*

3.2.10. План и поддерживающее его Руководство по документированию проекта жизненного цикла программного средства:

-общая структура комплекта документов;

-перечень исходных стандартов и нормативных документов;

-номенклатура и структура содержания каждого документа;

-требования к качеству, оформлению и обозначению исходных и результирующих документов;

-оценка соответствия стандартам установленного комплекта документов;

-регламент комплектования и хранения документов;

- правила обращения, изменения и сопровождения документов;
- план и графики подготовки, проверки, редактирования, согласования, утверждения и распространения документов для каждого этапа жизненного цикла:
 - * *исходные данные, требующиеся для успешного выполнения конкретного этапа документирования;*
 - * *контролируемые и документируемые данные о состоянии компонента и процесса разработки, регистрируемые после завершения этапа проекта;*
 - * *содержание процедур контроля состояния и качества компонентов и проекта ПС в целом и документов в процессе выполнения работ этапов;*
 - * *критерии оценки результатов выполненных работ и качества отчетных документов при завершении этапа и проекта ПС в целом;*
 - * *состав и содержание отчетных документов, представляемых для оценки состояния проекта, результатов завершенного этапа и работ и для использования на следующем этапе или при завершении проекта ПС;*
- обязанности и ответственность специалистов за содержание и качество конкретных документов;
- необходимые и используемые ресурсы, обеспечивающие создание конкретных документов требуемого качества.

3.2.11. Ведомость предварительного или детального проекта программного средства:

- согласованный перечень документов предварительного (детального) проекта программного средства,

	<p>предъявляемых заказчику на бумаге и/или в машинных файлах;</p> <p>-оценка соответствия комплекта, содержания и качества документов проекта требованиям спецификаций, технического задания и/или контракта (договора) на проектирование по разделам проекта программного средства.</p>
1.3	<p>3.3. Документы процессов разработки и программирования компонентов программных средств</p> <p>3.3.1. План разработки компонентов программного средства:</p> <p>-описание целей, стандартов и модели жизненного цикла, которые должны быть использованы в процессах разработки компонентов ПС;</p> <p>-идентификация стандартов на разработку компонентов ПС:</p> <p>* <i>требований к компоненту ПС;</i></p> <p>* <i>на процесс проектирования компонента ПС;</i></p> <p>* <i>на кодирования компонентов ПС для данного проекта;</i></p> <p>* <i>ссылки на стандарты для ранее разработанных компонентов ПС, включая коммерчески доступные ПС;</i></p>

- описание процессов жизненного цикла ПС, которые должны быть использованы для формирования конкретного жизненного цикла компонента данного проекта, включая критерии перехода между процессами и компонентами ПС;
- обоснование выбора используемой среды разработки ПС в аппаратной и программной части, включая выбор:
 - * методов и средств разработки требований к компонентам ПС;
 - * методов и средств проектирования компонентов ПС;
 - * языков программирования компонентов ПС;
 - * средств кодирования, компиляторов, редакторов связей и загрузчиков;
- аппаратная поддержка для инструментальных средств программирования компонентов ПС;
- план-график разработке компонентов ПС по этапам проекта.

3.3.2. План обеспечения качества компонентов программного средства:

- методы, которые должны быть использованы для того, чтобы достичь требуемое качество компонента ПС;
- описание среды обеспечения качества, включая область действия, организационную ответственность, интерфейсы, стандарты, процедуры, инструментальные средства и методы;
- утверждение полномочий службы обеспечения качества, её ответственности и независимости, включая

- полномочия специалистов на утверждение версий программных средств и компонентов;
- перечень и план-график работ обеспечения качества, которые должны быть выполнены для каждого процесса жизненного цикла компонента ПС и на протяжении всего жизненного цикла, включая:
- * *методы обеспечения качества, тестирование, просмотры, аудиты, отчетность, инспекции и мониторинг компонентов и процессов жизненного цикла ПС;*
- * *работы, связанные с отчетностью о дефектах, трассируемостью и системой корректирующих действий компонентов ПС;*
- синхронизация работ процесса обеспечения качества компонентов, относительно работ других процессов жизненного цикла ПС;
- определение отчетов, которые будут подготовлены в процессах обеспечения качества компонентов ПС.

3.3.3. Стандарты кодирования компонентов программного средства:

- методы, правила и инструментальные средства, которые будут использованы для кодирования компонентов ПС;
- перечень языков программирования и/или какое-либо их подмножество;
- * *ссылки на документы, инструкции и руководства которые однозначно определяют синтаксис, режим контроля, характер данных и побочные эффекты языка программирования;*
- * *ограничения на использование некоторых функций и возможностей языка программирования;*

* условия и ограничения, налагаемые на установленные соглашения кодирования, сложность логических или числовых выражений, а также обоснования для их использования;

* ограничения на использование инструментальных средств кодирования;

-стандарты представления, комментирования, оформления и документирования исходного кода, истории изменений, входные и выходные данные, а также наиболее значимые глобальные данные;

-соглашения по наименованию для компонентов, подпрограмм, переменных, констант, версий;

3.3.4. Руководство по программированию компонентов проекта комплекса программ:

-описание среды программирования:

* конфигурация и перечень компонентов системы, рабочие характеристики, возможности и ограничения, включая машинный цикл,

* длина слова, объем памяти и ее характеристики;

* перечень команд, прерываний, режимов работы;

* рабочие регистры, характеристики ввода/вывода;

* описание носителей данных;

-информация о возможностях применения языка программирования:

- * представление данных;
- * формат команд и методы адресации;
- * команды передачи управления;
- * процедуры и подпрограммы;
- * обработка прерываний;
- * синхронизация и таймеры;
- * возможности защиты памяти;
- * детальное описание каждой команды, их использование, синтаксис, время выполнения;
- * программирование управления вводом/выводом.

3.3.5. Документация на разработанный функциональный программный компонент или модуль программного средства:

- идентификатор, техническое задание и/или спецификация требований на разработку компонента; включая, имя автора и дату создания версии компонента;
- текст компонента ПС, написанный на исходном языке программирования, и команды компилятора, генерирующие объектный код из исходного текста, информация для редактирования связей и загрузки, а также текст комментариев;

- код, который является непосредственно пригодным для использования процессором объектного компьютера и загружаемый в аппаратные средства или вычислительную систему;
- описание программы в виде печатного документа и/или машинных файлов в составе:
 - * *краткая аннотация*;
 - * *описание решаемых задач*;
 - * *описание структуры и функций программного компонента*;
 - * *описание алгоритма компонента*;
 - * *описание и схема иерархии модулей сложного программного компонента*;
 - * *описание межмодульных интерфейсов программного компонента*;
 - * *описание пользовательских интерфейсов компонента*;
 - * *описания входных и результирующих данных компонента*;
 - * *описание программной и информационной среды функционирования компонента*;
 - * *описание способов проверки работоспособности и качества программного компонента и тестовые задачи*;
- фрагмент руководства пользователя – описание применения программного компонента (при необходимости);

	<ul style="list-style-type: none"> -исходный текст программы на языке программирования в виде печатного документа и/или машинных файлов с комментариями; -исходный текст и исполняемый объектный код программы на машинных носителях.
1.4	<p>3.4. Документы верификации и тестирования компонентов программных средств</p> <p>3.4.1. Состав базовых документов, регламентирующих верификацию и тестирование программных компонентов:</p> <ul style="list-style-type: none"> -руководство программистам по применению методологии, средств автоматизации и стандартов программирования при разработке компонентов ПС; -руководство программистам по управлению качеством компонентов и комплекса программ; -план обеспечения процессов верификации и тестирования средствами генерации тестов и обработки результатов функционирования компонентов ПС; -исходные тексты запрограммированных и оформленных компонентов и описаний данных; -общий план организации и порядка тестирования компонентов;

- описание оформления типовых тестов, генераторов тестовых данных и сценариев, используемых при тестировании компонентов;
- типовые формы (шаблоны) отчетов о результатах верификации и тестирования, достигнутых показателях качества, откорректированных программистами после завершения разработки компонентов;
- образцы текстов компонентов на языке программирования и в объектном коде реализующей ЭВМ после завершения верификации и тестирования.

3.4.2. Исходные данные для верификации программных компонентов:

- требования к функциональности, эффективности и к качеству системы, детализированные в исходных требованиях высокого уровня к ПС, производные требования к компонентам и обоснование их необходимости;
- каждое требование высокого уровня к ПС трассировано точными, однозначными и достаточно детализированными требованиями к компонентам, и они не конфликтуют друг с другом;
- отсутствуют неоднозначности и конфликты между требованиями к компонентам и к возможностями аппаратных и программных средств объектного вычислителя, особенно такими, как время реакции системы и характеристики аппаратуры ввода/вывода;
- оформление требований к ПС и компонентам полностью соответствует стандартам на создание спецификаций требований и любые отклонения от стандартов обоснованы;
- функциональные и конструктивные характеристики качества компонентов, предназначенные для реализации, полностью отражены и адекватны требованиям высокого уровня к ПС.

3.4.3. Результаты верификации корректности взаимодействия компонентов в составе программного средства:

- удостоверение внутренней непротиворечивости и полноты реализации компонентами требований к программному средству, посредством последовательного прослеживания выполнения комбинации из просмотров, анализов, разработки тестовых сценариев и процедур;
- требования к системе, предназначенные для программной реализации, корректно переработаны в спецификацию требований высокого уровня к комплексу программ, удовлетворяют исходным системным требованиям:
 - * *требования высокого уровня правильно переработаны в архитектуру ПС и в спецификации требований к функциональным компонентам низкого уровня, которые удовлетворяют требованиям высокого уровня;*
 - * *спецификации требований к функциональным компонентам ПС, расположенным между компонентами высокого и низкого уровня, каждый раз удовлетворяют требованиям более высокого уровня;*
 - * *архитектура ПС и спецификации требований к компонентам низкого уровня корректно переработаны, в удовлетворяющие им, исходные тексты программных и информационных модулей;*
 - * *исполняемый объектный код удовлетворяет требованиям к исходному тексту программных компонентов;*
- утверждена организационная ответственность внутри процесса верификации компонентов ПС и интерфейсы с другими процессами жизненного цикла;
- формализованы методы, которые будут использованы на каждом этапе процесса верификации ПС:
 - * *методы просмотра и трассирования спецификаций требований по уровням детализации описаний*

компонентов;

* *формализованы методы проверки трассируемости и оценки полноты покрытия верификацией компонентов ПС;*

-утверждены методики и процедуры выполнения просмотров и анализа, детализация верификации компонентов ПС в части области действия, глубины методов просмотров и анализа.

3.4.4. Исходные данные для тестирования компонентов:

-документация на разрабатываемый программный компонент:

* *техническое задание и/или спецификация требований на разработку программного компонента;*

* *описание программы в виде печатного документа и файла;*

* *описание функционирования и применения компонента в комплексе программ;*

* *исходный текст программы в виде печатного документа и файла;*

- выбранные методы верификации и тестирования компонентов:

* *статические или динамические;*

* *детерминированные или стохастические;*

- правила построения и описания тестов компонентов на разных уровнях их представления;

- правила структурного построения и интерфейсов компонентов между собой, а также с внешней средой и с пользователями;
- эталонные значения и/или распределения исходных и результирующих данных, отражающие требующиеся функции, сценарии тестирования и покрытие тестами компонента во всем заданном разнообразии его поведения;
- тестовые сценарии исходных и результирующих данных, являющиеся достаточно представительной выборкой из полной совокупности значений и распределений эталонных данных, ограниченной доступными ресурсами на тестирование компонента;
- доступные ресурсы на тестирование компонента:
 - * *доступные финансовые, трудовые и временные затраты на тестирование компонента;*
 - * *оснащенность процесса тестирования компонента программными и аппаратными средствами автоматизации;*
 - * *доступный состав и квалификация специалистов;*
- критерии качества тестирования и требуемая полнота покрытия тестами компонентов, в которые входят описания реализуемых функций и характеристики качества реализации функций;
- допуски на отклонение результатов функционирования программ и показателей качества компонентов от требуемых эталонных значений и распределений, в пределах которых считается, что компонент удовлетворяет предъявленным требованиям к качеству.

3.4.5. Организация, подготовка тестирования и обеспечение качества компонентов:

- системные и функциональные требования к конкретному компоненту, ранжированные требуемые показатели качества к каждому компоненту;
- декомпозиция обобщенных показателей качества ПС и компонентов по контролируемым этапам разработки и разделы по качеству в спецификациях требований на модули и компоненты, а также на их связь с методами и этапами тестирования;
- методы, технология и средства автоматизации разработки и тестирования, обеспечивающие создание каждого компонента с заданными показателями качества;
- методы и средства объективного измерения требуемого покрытия тестами и достигнутого качества каждого компонента на фиксированных этапах его разработки;
- документы и методики для обеспечения и контроля соблюдения правил и технологии проектирования, тестирования и обеспечения всего жизненного цикла компонентов и программного средства;
- структура и содержание (шаблон) каждого документа отражающего результаты этапа тестирования компонента, качество выполненных работ и полноты покрытия тестами.

3.4.6. Сценарии тестирования и спецификации тестов для каждого компонента:

- метод и вид тестирования адекватный компоненту, а также основной цели его выполнения;
- план тестирования в соответствии с выбранным методом с учетом ограниченных ресурсов испытаний, имеющихся для достижения заданной достоверности проверки качества компонента;

-спецификация общей схемы сценариев тестирования компонента:

* *характеристики компонента, требуемая полнота покрытия тестами для их контроля, охватываемые этой схемой и соответствующими ей тестами;*

* *специфические критерии для характеристик программного продукта, позволяющие оценивать компонент: годен-не годен;*

-описания контрольных сценариев тестирования – набор конкретных тестовых значений и соответствующих им эталонов:

* *действительные значения, используемые в качестве исходных тестов;*

* *ожидаемые, эталонные выходные значения результатов тестирования;*

* *ограничения по процедурам тестирования для каждого конкретного сценария;*

-спецификация конкретного сценария тестирования, которая идентифицирует все этапы, требуемые для работы системы в целом или компонента, и для выполнения контрольных сценариев, чтобы реализовать связанные с ними тесты;

-задание на верификацию и тестирование с указанием контролируемых параметров, исходных данных и результирующих эталонов;

-результаты функционирования компонента тестирования при подготовленных тестах и заданиях;

-сравнения результатов тестирования с эталонами и обнаруженные отклонения (дефектов или ошибок) для

- проведения дополнительного тестирования с целью диагностики и локализации дефектов;
- оценки полноты проведенного тестирования, степени покрытия компонента тестами выбранным методом и необходимости применения других методов и видов тестирования для увеличения покрытия тестами;
- оценка характеристик качества компонента, исходных и выходных данных в спецификациях и сценариях;
- оценки наличия ресурсов для продолжения тестирования и момента его завершения, а также для определения достигнутых характеристик качества компонента или выбора очередного метода тестирования;
- документы результатов процедур верификации и тестирования программы.

3.4.7. План тестирования программного компонента:

- цель, вид и назначение плана тестирования конкретного компонента;
 - модули и компоненты, подлежащие тестированию;
 - назначение каждого тестового сценария, набор входных данных, условия исполнения тестов, ожидаемые результаты, требуемые критерии покрытия и критерии полноты выполнения тестов;
 - пошаговые инструкции того, как каждый тестовый сценарий должен быть инициирован и выполнен, как должны быть оценены результаты тестирования и какая среда тестирования должна быть использована;
 - организация работ, основной график их выполнения:
- * *задачи исполнения каждого теста и сценарий проверки;*

- * методы и критерии оценки качества тестирования;
- * входные и результирующие данные тестирования;
- * используемые ресурсы сценария тестирования;
- используемые технологические средства и методики тестирования;
- номенклатура и содержание оформляемых отчетных документов;
- взаимодействие плана тестирования компонентов с планами обеспечения качества и с управлением конфигурацией и корректировками компонентов и ПС в целом;
- детализация и документирование процесса тестирования на каждом этапе ЖЦ ПС:
- * графики решения частных задач тестирования компонентов;
- * оценки результатов тестирования и риска качества компонентов;
- * распределение ответственности между специалистами за качество результатов тестирования и достигнутые характеристики компонентов.

3.4. 8. Отчет о результатах верификации и тестирования компонентов:

- справка о передаче компонента на тестирование, когда в разработке участвуют независимые группы

программистов и тестировщиков;

-журнал выполнения плана тестирования, используемый бригадой тестировщиков компонентов для регистрации:

* *сценариев и операций, которые имели место во время выполнения тестирования компонента;*

* *аномальных событий и дефектов для диагностики и локализации причин аномалий;*

* *изменений и корректировок, которые произведены в компонентах;*

-результаты верификации и тестирования:

* *результат выполнения (прошел/не прошел) для каждого просмотра, анализа и выполненного теста и заключительный результат верификации и тестирования компонента;*

* *элемент конфигурации и/или версия ПС, которые прошли просмотр, анализ или верификацию соответствующих компонентов;*

* *результаты оценивания покрытия и анализа трассируемости для выполнения тестов, просмотров и анализов в процессе верификации и тестирования;*

-итоговый обобщенный отчет верификации и тестирования компонентов ПС:

* *результаты работ по верификации и тестированию, связанные с одной или более спецификациями сценариев или видов проверки компонентов и ПС в целом;*

- * отчеты о факультативных исследованиях ПС и системы в целом, выполненных испытателями или пользователями за пределами спецификаций требований и документации;
- * компоненты комплекса программ выполняют все (частично) декларируемые в документации функции;
- * полученные выходные данные находятся в допустимых пределах отклонений от эталонных, заданных в требованиях на разработку комплекса программ или спецификациях на компоненты;
- * при обработке тестовых данных не выявлено ошибок или отклонений функционирования от описанного в программной документации;
- * программная документация соответствует требованиям стандартов.

3.4.9. Методика комплексирования функциональных компонентов:

- тестирование идентичность исходного текста функционального компонента, представленного на носителе данных, с исходным текстом, представленным в программном документе на крупную функциональную задачу;
- комплексирование в статике модулей и компонентов, входящих в функциональный компонент, проверка интерфейсов между модулями и выявлены их нестыковки с описаниями в спецификациях на функциональный компонент;
- устранение невязок интерфейсов между модулями и компонентами, входящими в функциональный компонент;
- анализ потоков управления и установления степени покрытия тестами графовой модели функционального

компонента при тестировании программы по выделенным маршрутам функционального компонента:

- * установить наличие или отсутствие тупиковых ветвей в маршрутах;
- * установить наличие хотя бы одного маршрута исполнения программы для каждой из заданных в спецификации функции компонента;
- * выделить имеющиеся на маршрутах циклы функционального компонента и определить порядок их тестирования;
- * используя логические условия в вершинах графа, описывающего потоки управления функционального компонента, сформировать исходные данные для создания тестовых наборов входных данных для проверки каждого выделенного маршрута;
- анализ потоков данных – установить связь наборов входных переменных программы и выходных, участвующих в исполнении функционального компонента по определенному маршруту, соответствие между областями определения наборов данных (входных и выходных) и маршрутами их обработки в функциональном компоненте;
- оценка результатов тестирования, качества и корректности конкретного функционального компонента статике без взаимодействия с другими функциональными компонентами;
- * функциональный компонент имеет корректную структуру;
- * для каждой из функций компонента существует непустое множество маршрутов ее реализации;
- * каждый из функциональных маршрутов завершается за конечное число шагов;

* не обнаружено нереализованных и тупиковых функциональных маршрутов;

* программная документация соответствует требованиям стандартов и реализуемым функциям компонента;

-подготовка тестов и сценариев для тестирования функционального компонента в статике, во взаимодействии с другими функциональными компонентами при некоторых постоянных значениях времени в соответствии с заданной временной диаграммой комплекса программ;

-комплексирование взаимодействия функционального компонента с другими компонентами и с базой данных вне реального времени;

-подключение функционального компонента к операционной системе в реализующей ЭВМ и оценка возможности достаточно полного решения соответствующих функциональных задач в статике при постоянных значениях реального времени;

-завершение тестирования функционального компонента с другими группами программ, наращивание состава компонентов до полного комплекта реализуемых функциональных задач в ПС;

-обработка результатов отладки, оценка качества и корректности функционального компонента во взаимодействии с другими группами в статике, вывод о достаточности данного набора тестов полностью проверить декларируемые в документах функциональные возможности;

-подготовка тестов для тестирования функционального компонента в реальном времени, сценарии тестирования и диаграммы изменения реального времени, которым соответствуют определенные содержания тестов в имитируемой внешней среде;

- завершение тестирования качества функционального компонента в реальном времени, программ ввода-вывода и взаимодействия с внешней средой, визуализации и организации всего вычислительного процесса в реальном времени;
- обработка результатов тестирования и оценка качества функционального компонента в реальном времени, в полном его окружении с другими функциональными компонентами, удовлетворяющими критериям качества, передача от его разработчиков на комплексирование специалистам, ответственным за тестирование комплекса программ в целом.

3.4.10. Оценка реализации комплексирования функциональных компонентов комплексов программ:

- удостоверение качества функционирования компонентов в соответствии с спецификациями требований, а также в критических ситуациях по условиям и логике решения задач (стрессовое тестирование) для испытаний исполнения функциональных компонентов в нештатных ситуациях;
- определение корректности использования ресурсов памяти и производительности вычислительной системы при оценке качества и безопасности исполнения функциональных компонентов при ограниченности ресурсов ЭВМ;
- оценка эффективность защиты функционального компонента от искажений исходных данных для выявления дефектов, проявляющихся при ложных или искаженных входных данных;
- оценки эффективности защиты от сбоев аппаратуры и не выявленных ошибок программ и данных для проверки качества средств контроля и оперативного восстановления (рестарта) при различных, непреднамеренных искажениях функционирования компонентов ПС;
- измерение достигнутых значений надежности и безопасности базовых версий функциональных компонентов

- ПС и определение основных характеристик качества комплекса программ;
- оценка удобства эксплуатации и взаимодействия оператора-пользователя с комплексом программ для обнаружения дефектов представления и отображения исходных и результирующих данных;
 - оценка удобства и качества инсталляции и подготовки пользовательских версий комплекса программ для выявления дефектов методов и средств адаптации функциональных компонентов базовых версий к параметрам среды и конкретным условиям применения у пользователей;
 - оценка возможности и качества функционирования компонентов базовых версий комплекса программ, при реконфигурациях оборудования для обнаружения дефектов, проявляющихся при изменении состава или характеристик компонентов аппаратуры вычислительной системы или внешней среды;
 - оценка готовности функциональных компонентов комплекса программ к сопровождению и развитию версий путем настройки базовых версий на условия конкретного применения у пользователей, анализ удобства модификации и корректировки версий ПС;
 - контроль полноты и корректности документации, обнаружение и устранение ошибок несоответствия функциональных компонентов реального комплекса программы в ЭВМ, сопровождающей его технологической и эксплуатационной документации.

1.5	3.5. Документы квалификационного тестирования, испытаний и оценивания качества программных средств
-----	---

3.5.1. Методика генерации тестов имитирующих внешнюю среду и обработку результатов квалификационного тестирования:

-выбор и установление статуса испытания ПС и соответствующих тестов:

- * *ординарные испытания, которым подвергается широкий спектр ПС с относительно невысокими требованиями к качеству, которые предстоит эксплуатировать в не критических системах;*
- * *аттестационные испытания, которым подвергаются ответственные категории ПС, чьи ошибки могут нанести большой ущерб;*
- * *сертификационные испытания критических ПС, эксплуатация которых недопустима без высоких гарантий качества, удостоверяемых уполномоченным государственным или ведомственным органом;*

-имитация конкретных тестов с реальными характеристиками, адекватными объектам внешней среды, с учетом:

- * *статуса испытаний;*
- * *глубины знаний об алгоритмах функционирования объектов внешней среды;*
- * *характеристик компонентов внешней среды;*

-выбор средств имитации внешней среды, обеспечивающих реальный масштаб времени при тестировании и включающих:

- * *аналоги объектов внешней среды для генерации тестов, представляющих коррелированные логические*

переменные, которые трудно описать и смоделировать на ЭВМ;

** данные с рабочих мест операторов-пользователей с учетом особенностей и квалификации человека, которому предстоит использовать испытываемые программы в реальной системе обработки информации;*

** данные натурных экспериментов с объектами внешней среды отражающие характеристики и динамику функционирования объектов, которые трудно или опасно подключать для непосредственного взаимодействия с недостаточно проверенным ПС и для аттестации адекватности имитаторов некоторых объектов внешней среды;*

** имитацию эталонных характеристик объектов внешней среды в идеальных условиях - при отсутствии искажений исходных данных, ошибок в измерениях их параметров, сбоев и предумышленных отказов;*

-математическое моделирование внешней среды - поддержка процесса тестирования с помощью имитации данных из внешних для ПС компонентов системы, что позволяет:

** проводить длительное непрерывное генерирование имитируемых данных для определения характеристик качества функционирования ПС в широком диапазоне изменения условий и параметров, что зачастую невозможно при использовании реальных объектов;*

** расширять диапазоны характеристик имитируемых объектов за пределы реально существующих или доступных источников данных, а также генерировать потоки информации, отражающие перспективные характеристики создаваемых информационных систем и объектов внешней среды;*

** создавать тестовые данные, соответствующие критическим или опасным ситуациям функционирования объектов внешней среды, которые невозможно или рискованно реализовать при натурных экспериментах;*

- * обеспечивать высокую повторяемость имитируемых данных при заданных условиях их генерации и возможность прекращения или приостановки имитации на любых фазах моделирования внешней среды;
- регистрация характеристик тестовых данных на соответствие заданным обобщенным характеристикам каждого объекта внешней среды и исходным данным сеанса испытаний:
 - * данные, характеризующие исходную для испытаний тестовую информацию;
 - * выходные эталонные результаты тестирования;
 - * маршруты исполнения программных компонентов и их операторы при некоторых фиксированных тестовых данных;
 - * аномальные события, сбои, отказы и данные, характеризующиеся отклонением результатов тестирования от эталонов за допустимые пределы и ограничения;
 - * характеристики превышения допустимого использования различных ресурсов объектной ЭВМ;
- результаты оперативной обработки итогов испытаний тестируемого комплекса реального времени, по упрощенным алгоритмам с большой пропускной способностью, обеспечивающим сохранение реального масштаба времени для испытываемого ПС;
- результаты обобщающей обработки накопленных результатов испытаний вне реального времени после завершения одного или серии испытаний характеристик качества функционирования ПС.

3.5.2. Методика применения проблемно-ориентированной системы квалификационного тестирования и испытаний комплексов программ:

- выделение требований и характеристик, подлежащих проверке при испытаниях функциональных компонентов и ПС в целом, обеспечивающих контроль проектных решений;
- выявления причин сбоев и отказов функционирования компонентов и ПС в целом;
- определения реальных характеристик качества функционирования системы и комплекса программ;
- проверка соответствия функционирования ПС и системы требованиям спецификаций и технического задания;
- установления необходимой продолжительности и режимов испытаний для достаточной проверки выполнения требований технического задания и соответствия предъявленной документации;
- определение тестов и реализация процесса тестирования разработчиком: ввод тестовых наборов; генерация тестовых данных; ввод ожидаемых, эталонных результатов;
- выполнение участка тестируемой программы между контрольными точками, для которого вводимые данные могут быть отредактированы и включены в последующие тестовые сценарии;
- анализ и обработка тестовых результатов - использование возможности средства тестирования автоматически анализировать тестовые результаты: сравнение ожидаемых и реальных результатов; сравнение файлов; статистическую обработку результатов;
- анализ покрытия тестами объектного кода комплекса программ для обнаружения: операторов, которые не были выполнены; процедур, которые не были вызваны; переменных, к которым не были обращения;
- анализ производительности при исполнении комплекса программ: загрузки центрального процессора;

загрузки памяти; обращения к специфицированным элементам данных или сегментам объектного кода; временные характеристики функционирования испытываемой программы.

3.5.3. Методика, содержание и сценарии квалификационного тестирования и испытаний программных средств:

-содержание тестовых сценариев и процедур, которые используют для выполнения квалификационного тестирования системы или функциональных компонентов ПС:

* *каждый тест должен иметь уникальный для проекта идентификатор и ссылку на соответствующий пункт в документе Программа квалификационного тестирования;*

* *инструкции для проведения тестирования, описание аппаратуры и ПС, а также инструкции для возможности повторного тестирования;*

* *ссылки на проверяемые требования спецификаций, условия выполнения (конфигурация аппаратуры и ПС), входные данные, ожидаемые результаты, критерии оценки результатов, процедуры тестирования для каждого тестового сценария, допущения и ограничения испытаний;*

-план квалификационного тестирования комплекса программ:

* *описание тестовой среды, которая будет использована при тестировании, содержание выполняемых тестов и план-график реализации тестирования;*

* *идентификация, перечень и используемые виды аппаратных средств, интерфейсного оборудования, устройств связи, дополнительных внешних устройств, генераторов тестовых данных, устройств синхронизации тестов;*

- * организации, принимающие участие в квалификационном тестировании ПС, их роли и ответственность;
 - * план-график тестирования и матрица трассирования квалификационных тестов с отношением к конкретным требованиям спецификаций компонентов и ПС;
- квалификационное тестирование программного средства вне системы:
- * испытания выполнения всех требований контракта и спецификаций к характеристикам качества комплекса программ;
 - * подготовка к интеграции комплекса программ и аппаратуры системы;
 - * оценка достигнутых характеристик качества и возможности автономного применения программного средства по назначению;
- интеграция и тестирование комплекса программ в составе аппаратуры системы:
- * испытания внешних и внутренних интерфейсов комплекса программ на соответствие требованиям к системе;
 - * оценка реализуемости и планирование испытаний комплекса программ в составе системы;
 - * анализ полноты и корректности документации на комплекс программ в составе документации системы;
- квалификационное тестирование характеристик качества системы с данным комплексом программ:
- * установление соответствия достигнутых характеристик качества ПС и системы требованиям контракта

и спецификаций;

** удостоверение адекватности и качества технологической и эксплуатационной документации на систему и ПС результатам квалификационного тестирования;*

-предварительные испытания главного конструктора – разработчика, которые могут совмещаться с завершением квалификационного тестирования функциональных компонентов, оформляются документально и являются основанием для предъявления заказчику на завершающие совместные, квалификационные испытания программного продукта;

-опытная эксплуатация системы и комплекса программ разработчиками с участием испытателей и некоторых пользователей, назначаемых заказчиком (результаты могут учитываться при проведении совместных испытаний для их сокращения);

-совместные, квалификационные испытания системы и комплекса программ комиссией заказчика и разработчика, в которой участвует главный конструктор разработки и некоторые ведущие разработчики, или аттестованной сертификационной лабораторией:

-исходные документы комиссии при испытаниях:

** утвержденное заказчиком и согласованное с разработчиком техническое задание и спецификации требований на комплекс программ;*

** действующие государственные и ведомственные стандарты на проектирование и испытания комплекса программ и на техническую документацию;*

** Программа испытаний по всем требованиям спецификаций и технического задания;*

* методики испытаний по каждому разделу Программы, требований спецификаций и технического задания на проект;

* комплект сопроводительной технологической и эксплуатационной документации на комплекс программ;

-отчет о квалификационном тестировании (испытаниях) ПС, выполненном для системы и комплекса программ:

* общая оценка результатов квалификационного тестирования, идентификация всех дефектов, несоответствий и ограничений;

* описание возможных различий тестовой и эксплуатационной внешней среды и системы;

* описание рекомендуемых улучшений в тестируемом комплексе программ;

* детальные результаты квалификационного тестирования и испытаний;

* описание обнаруженных и устраниенных дефектов;

* оформленный акт о завершении работ и контракта на создание версии комплекса программ и системы.

3.5.4. Программа испытаний комплекса программ:

-перечень конкретных проверок (решаемых задач), которые следует осуществлять при испытаниях для подтверждения выполнения требований технического задания, спецификаций и нормативной документации, со ссылками на соответствующие разделы методик испытаний;

- объект испытаний (компонент или комплекс программ), его назначение и перечень основных документов, определивших его разработку;
- цель испытаний с указанием всех спецификаций требований, технического задания и характеристик качества ПС, подлежащих проверке;
- ограничения на проведение испытаний (экономические, технологические, временные, ресурсы специалистов);
- этапы жизненного цикла, на которых должно проводиться тестирование компонентов или ПС в целом;
- с собственно Программа испытаний комплекса программ:

- * *перечень конкретных проверок, которые следует осуществлять при испытаниях для подтверждения выполнения требований спецификаций, технического задания и нормативной документации, со ссылками на соответствующие разделы методик испытаний;*
- * *план тестирования для последовательной проверки по всем разделам технического задания и дополнительным требованиям спецификаций, formalизованным отдельными решениями разработчиков и заказчика;*
- * *сценарии проверок ПС соответствия техническому заданию и оценки степени выполнения требований функционального назначения системы и комплекса программ;*
- * *оценка комплектности, достаточности состава и качества документации на комплекс программ;*
- * *определение потребности в количестве и квалификации пользователей и обслуживающего персонала;*

-основы методик испытаний, однозначно определяющие:

- * *содержание проверяемых характеристик;*
- * *условия и сценарии тестирования функциональных компонентов и комплекса программ;*
- * *инструментальные средства, используемые для автоматизации испытаний;*
- * *содержание методик обработки и оценки результатов квалификационного тестирования по каждому разделу Программы испытаний.*

3.5.5. Методики проведения испытаний комплекса программ по отдельным характеристикам качества:

-объект испытаний – полное наименование комплекса программ, обозначение и комплектность испытываемого функционального компонента или ПС в целом;

-цель испытаний – конкретные цели и задачи, которые должны быть достигнуты и решены в процессах испытаний;

-общие положения методов испытаний:

- * *перечень руководящих документов, на основании которых проводят испытания;*
- * *место, внешняя среда и продолжительность испытаний;*
- * *идентификаторы организаций, участвующих в испытаниях;*

- * перечень математических и комплексных моделей, применяемых для имитации внешней среды и оценки характеристик ПС;
 - * перечень и результаты ранее проведенных испытаний;
 - * перечень предъявляемых на испытания комплектов документов, откорректированных по результатам ранее проведенных испытаний;
- размеры испытаний по разделам Программы:
- * перечень этапов тестирования, испытаний и оценок;
 - * количественные и качественные характеристики комплекса программ, подлежащие измерению и оценке;
 - * последовательность проведения этапов тестирования и режимы испытаний;
 - * перечень работ, проводимых после формального завершения испытаний и требования к ним;
 - * объем, методики и порядок обработки результатов испытаний;
- условия и порядок проведения испытаний:
- * условия начала и завершения отдельных этапов испытаний функциональных компонентов и комплекса программ;
 - * имеющиеся ограничения в условиях проведения этапов испытаний;

- * требования к техническому обслуживанию системы в процессе испытаний;
 - * необходимые меры, обеспечивающие безопасность и безаварийность проведения испытаний системы и ПС;
 - * порядок взаимодействия и ответственность организаций и подразделений, участвующих в испытаниях за корректность и безопасность;
 - * конкретное распределение функциональных задач и обязанностей организаций, участвующих в испытаниях за результаты;
 - * требования к квалификации персонала, проводящего испытания, и порядок его допуска к испытаниям;
- материально-техническое и стендовое обеспечение испытаний комплекса программ;
- метрологическое обеспечение корректности испытаний с распределением задач и ответственности организаций и специалистов, участвующих в испытаниях, за выполнение соответствующих этапов, мероприятий и оценку характеристик комплекса программ;
- отчетность и документирование результатов испытаний:
- * перечень отчетных документов, которые должны оформляться в процессе испытаний и по их завершению;
 - * перечень организаций разрабатывающих, согласующих и утверждающих конкретные отчетные документы;
 - * сроки оформления и утверждения отчетных документов испытаний;
- форма и содержание акта и итогового отчета о результатах испытаний;

-акт содержания технического состояния ПС и системы после испытаний.

3.5.6. Протоколы по результатам испытаний функциональных компонентов и/или комплекса программ:

-идентификатор объекта испытаний;

-цель испытаний;

-назначение квалификационного тестирования и разделы требований спецификации, технического задания и Программы, по которым проводились испытания;

-список должностных лиц и их квалификации, проводивших испытания;

-перечень пунктов Программы испытаний, по которым успешно проведены испытания;

-перечень методик, в соответствие с которыми успешно проведены испытания, обработка и оценка результатов;

-условия и сценарии проведения квалификационного тестирования и характеристики исходных данных;

-отчет об отказах, сбоях и аварийных ситуациях, выявленных при испытаниях;

-отчет о корректировках параметров среды и объекта испытаний, комплекса программ и технической документации;

-обобщенные результаты квалификационных испытаний с оценкой их на соответствие спецификациям требований технического задания и другим руководящим документам, а также технической документации;

- протоколы просмотров и аудитов, протоколы совещаний, регистрация отклонений от санкционированных процессов и протоколы проверки соответствия комплекса программ требованиям спецификаций;
- выводы о результатах испытаний и соответствии созданного комплекса программ и/или функционального компонента определенному разделу требований спецификаций и технического задания.

3.5.7. Итоговый отчет результатов разработки программного средства:

- краткий обзор системы, включая описание ее функций и их распределение на программную и аппаратную реализацию, архитектура, используемые процессоры, интерфейсы аппаратных и программных средств, требования по обеспечению безопасности;
- фактически использованная модель жизненного цикла ПС;
- ссылки на документы жизненного цикла ПС, являющиеся выходными результатами процессов разработки и интегральных процессов комплекса программ;
- функции ПС с акцентированием на обеспечение характеристик качества, безопасности и используемую концепцию разбиения компонентов ПС;
- основные характеристики ПС – размер исполняемого объектного кода, ограничения по времени и памяти, ограничения ресурсов разработки, значения и способы измерения каждой характеристики;
- связь между представляемыми документами комплекса программ и документами, определяющими систему, а также способы передачи комплекса документов жизненного цикла ПС заказчику;
- идентификация конфигурации ПС, посредством указания регистрационного номера и версии;

- обзор изменений и корректировок в ЖЦ ПС с указанием изменений, вызванных отказами, влияющими на качество и безопасность, с идентификацией изменений, выполненных после сдачи предыдущей версии;
- перечень сообщений о дефектах комплекса программ, не устраниенных ко времени завершения испытаний, включая заявления о функциональных ограничениях ПС;
- утверждение о соответствии проекта требованиям стандартов и обзор методов, позволяющих доказать выполнение критериев, определенных в исходных данных проекта и планах ПС.

3.5.8. Акт завершения работ по проекту программного средства:

- идентификатор проекта и завершенной работы;
- список представителей организации-разработчика и организации-заказчика, составивших акт;
- дата завершения проекта и работ;
- перечень и наименования документов, на основании которых осуществлен проект и проводилась работа;
- перечень и наименования документов, содержащих обобщенные результаты выполненного проекта ПС;
- основные результаты завершенного проекта ПС;
- заключение о результатах завершенного проекта и степени выполнения технического задания и спецификаций требований к ПС с положительным или отрицательным итогом;
- рекомендации по развитию и внедрению результатов проекта ПС;

-приложения:

* *комплект технологической и эксплуатационной документации на комплекс программ;*

* *исходные документы на разработку проекта ПС;*

* *полный отчет о результатах квалификационных испытаний ПС.*

3.5.9. Акт приемки программного средства в промышленную эксплуатацию:

-идентификатор системы и/или ПС, принимаемых в эксплуатацию;

-сведения о статусе приемочной комиссии (государственная, межведомственная, ведомственная), ее составе и основании для работы;

-период времени работы комиссии приемки в эксплуатацию;

-идентификаторы организации-разработчика, организации-соисполнителя и организации заказчика;

-перечень и наименования исходных документов, на основании которых разработано ПС;

-состав и описание функций ПС, принимаемых в эксплуатацию;

-перечень и описание компонентов технического, программного, информационного и организационного обеспечения, принимаемых в эксплуатацию;

-перечень и наименования комплекса документов, предъявленных комиссии;

- заключение о результатах опытной эксплуатации ПС;
 - оценка соответствия принимаемого ПС техническому заданию, спецификации требований и контракту на его создание;
 - краткая характеристика и основные результаты выполненной работы по созданию ПС;
- * *оценка научно-технического уровня комплекса программ;*
- * *оценка экономической эффективности от возможного внедрения комплекса программ;*
- решение комиссии о возможности принятия ПС в промышленную эксплуатацию;
 - рекомендации комиссии по дальнейшему развитию системы и комплекса программ;
 - приложения:
- * *Программа и протоколы испытаний;*
- * *протоколы заседаний комиссии;*
- * *перечень технических средств и характеристик внешней среды, которые использовала комиссия при испытаниях и приемке ПС;*
- * *справка о применении в ПС нормативных документов и унифицированных форм (шаблонов) документов.*

1.6

3.6. Документы сопровождения и конфигурационного управления версиями программного средства

3.6.1. Описание среды жизненного цикла и конфигурации программного средства:

- аппаратная и программная среда разработки, генерации, повторной верификации или модификации ПС на протяжении всего жизненного цикла;
- инструментальные средства разработки ПС: компиляторы, редакторы связей и загрузчики, средства обеспечения целостности данных;
- среда и методики верификации и тестирования компонентов, используемые при корректировке программного средства;
- аттестованные инструментальные средства обеспечения и корректировки жизненного цикла ПС и соответствующая документация об аттестации этих средств;
- идентификаторы конфигурации функциональных компонентов и программного средства; исходные тексты программ и исполняемый объектный код;
- ранее разработанные компоненты ПС, если они используются в данном программном средстве;
- комплекс технологических документов обеспечения жизненного цикла ПС;
- инструкции для компоновки исполняемого объектного кода, инструкции для компилирования и

- редактирования связей; процедуры, используемые для восстановления при регенерации, тестировании или модификации компонентов и ПС;
- контроль целостности данных для исполняемого объектного кода, документы управления конфигурацией, генерируемые в процессе управления конфигурацией, включая отчеты управления конфигурацией, указатели состояния конфигурации ПС и среды жизненного цикла комплекса программ.

3.6.2. План управления конфигурацией программного средства:

- описание среды управления конфигурацией, включая процедуры плана графика, инструментальные средства, методы, стандарты, организационную ответственность и интерфейсы специалистов;
- описание процессов управления конфигурацией в жизненном цикле ПС, которые обеспечивают реализацию целей данного процесса;
- компоненты изменения конфигурации, которые должны быть идентифицированы, методы идентификации документов жизненного цикла ПС, связь идентификации компонентов, ПС и системы;
- содержание и идентификация сообщений об изменениях ПС, процедуры регистрации сообщений о дефектах и взаимодействие отчетов о дефектах и контроле изменений;
- контроль изменений и базовая версия, обеспечивающие целостность компонентов конфигурации и базовой версии ПС;
- методы оценки и определения приоритетов в устранении дефектов, утверждении изменений, реализации решений об изменениях и связь этих методов с отчетами о дефектах и контролем изменений;

- отчет о текущем состоянии конфигурации, определение места хранения информации, как она воспроизведена для отчетов и когда она будет доступна специалистам;
- архивирование, получение из архива и выпуск официальной базовой версии ПС, контроль целостности, способы внесения информации в архив и получения из архива, метод и полномочия для выпуска версии комплекса программ;
- контроль среды жизненного цикла, инstrumentальных средств для разработки, комплексирования, верификации, тестирования и загрузки ПС;
- контроль целостности технологических документов жизненного цикла ПС;
- определение состава документов жизненного цикла ПС, генерируемых в процессе управления конфигурацией, включая отчеты управления конфигурацией компонентов, указатели состояния конфигурации и среды жизненного цикла ПС.

3.6.3. Отчеты пользователей о выявленных дефектах и предложениях по корректировке комплекса программ:

- рекомендации пользователям по выявлению и регистрации условий проявления и содержания дефектов эксплуатируемых версий ПС;
- идентификация и регистрация аномального поведения ПС, несогласованности процессов с планами и стандартами разработки, недостатки документации жизненного цикла ПС;
- описание дефекта, достаточное для его понимания и устранения, описание возможных корректирующих действий, предназначенных для устранения зарегистрированного дефекта;

- идентификатор пользователя, представившего отчет о дефектах и/или предложениях;
- дата фиксирования дефекта или предложения на изменение ПС;
- номер и параметры адаптации пользовательской версии ПС, на которой обнаружен дефект;
- идентификация компонента конфигурации и/или этапа жизненного цикла ПС, где был обнаружен дефект;
- подробное описание сценария и исходных данных, при которых выявлен дефект и документы результатов его регистрации;
- предположение о причине, вызвавшей проявление дефекта;
- идентификация компонента конфигурации, который необходимо модифицировать, или описание процесса, который должен быть изменен;
- описание возможных корректирующих действий, предназначенных для устранения зарегистрированного дефекта;
- предложение по модификации ПС и его компонентов для устранения дефекта или совершенствования функционирования программ.

3.6.4. Описания выявленных дефектов и предложений по совершенствованию функций версии программного средства:

- отчеты пользователей о дефектах и предложениях в рубрикации раздела 6.3;

- идентификатор разработчика, которому передан отчет пользователя для анализа дефекта или предложения;
- дата начала анализа отчета пользователя;
- идентификатор сценария проявления повторяемости дефекта на пользовательской версии и необходимости дальнейшего анализа дефекта на базовой версии ПС;
- тесты, исходные данные и сценарий, при которых проявляется выявленный дефект;
- результаты анализа предложения на выполнение изменения, причин и источника выявленного дефекта;
- рекомендации о возможных способах устранения дефекта или о реализации предложения по совершенствованию программ и базы данных;
- оценки сложности, трудоемкости, эффективности и срочности модификации программ и базы данных;
- оценки влияния предлагаемых изменений на возможность эксплуатации версий ПС, имеющихся у пользователей.

3.6.5. Описания подготовленных и утвержденных корректировок и обобщенных характеристик новой базовой версии программного средства:

- идентификатор специалиста, который разработал модификацию компонентов программ и базы данных;
- дата разработки предлагаемой модификации;
- причина изменения программ и/или базы данных (дефект, совершенствование);

- содержание изменений программ и базы данных;
 - содержание изменений документации на компоненты или версию ПС;
 - корректность результатов тестирования базовой версии ПС с разработанными изменениями;
 - дата и ответственное лицо, утвердившее реализацию модификации версии ПС;
 - квалификация решения на изменение: частная модификация вследствие исправления дефекта или издание новой версии ПС;
 - результаты испытаний модификации и обобщенные характеристики базовой версии ПС после внесения изменений;
 - решение по распространению пользователям результатов проведенной модификации или новой версии ПС;
 - решение по целесообразности сохранения сопровождения предшествующих версий ПС;
 - адрес хранения корректировок, документов и квалификационных тестов выполненной модификации и/или новой базовой версии ПС;
- идентификация новой конфигурации, протоколы об установлении новой базовой версии и её регистрации в архиве;
- архивирование, получение из архива и выпуск официальной версии: контроль целостности базовой версии ПС, способы внесения информации в архив и получения из архива, метод и полномочия для выпуска версии комплекса программ; отчеты об истории выполнения изменений версий ПС; протоколы о передаче новой

версии ПС в архив;

-протоколы о выпуске новой версии ПС для применения пользователями.

3.6.6. Извещение пользователям о выпуске новой версии программного средства и/или о прекращении сопровождения предшествующей версии:

-краткое обоснование причин модификации или прекращения сопровождения версии ПС;

-описание содержания и характеристики основных изменений в новой версии ПС;

-рекомендации по корректировке, приобретению или замене пользовательской версии ПС.

3.6.7. Описание новой базовой версии программного средства:

-идентификация системы и комплекса программ, к которым применяется данный документ, включая регистрационные номера процедур утверждения конфигураций и номера базовых версий ПС;

-краткий обзор назначения и спецификации требований новой версии системы и ПС, особенности истории разработки, эксплуатации и сопровождения системы и комплекса программ, идентификация заказчика, пользователей, разработчика, а также организаций, осуществляющей сопровождение, регистрацию текущих и планируемых мест установки системы и ПС для пользователей;

-идентификация физических носителей, содержащих новую зарегистрированную и утвержденную базовую версию ПС и связанную с ней документацию;

-идентификация комплекта новых зарегистрированных, утвержденных компьютерных файлов, содержащих

базовую версию ПС;

- перечень всех изменений, внесенных в файлы и документы после выпуска предыдущей базовой версии ПС;
- полный комплект документов новой базовой версии ПС;
- инструкция по установке и инсталляции новой версии ПС.

3.6.8. План передачи и внедрения новой базовой версии программного средства пользователям:

- общий обзор результатов разработки, комплекта требований и особенностей новой базовой версии системы и ПС; заказчиков, пользователей, разработчиков и организаций, осуществляющих сопровождение, запланированные рабочие места и перечень передаваемых документов;
- контроль комплектности и состояния аппаратного и программного обеспечения, а также другие ресурсов, необходимых для поддержки всего жизненного цикла передаваемого нового комплекса программ;
- запланированные сроки установки новой версии ПС у определенных пользователей;
- комплект системы файлов и документов, относящихся к передаваемой новой базовой версии ПС;
- детальное описание ресурсов, необходимых для инсталляции передаваемого ПС, требования к квалификации и составу персонала, чтобы специфицировать, разработать, документировать, тестировать, оценивать, контролировать, копировать и распространять новую базовую версию ПС;
- перечень рекомендуемых мероприятий, в том числе консультации и лекции, которые должен проводить разработчик в целях поддержки применения передаваемого нового ПС;

- описание процесса подготовки персонала, который будет осуществлять поддержку передаваемого ПС: тематика, дата, продолжительность и место проведения теоретических и практических занятий;
- порядок внедрения, включающий в себя все работы, необходимые при передаче и инсталляции новой версии ПС со стороны организаций, осуществляющих сопровождение.

3.6.9. Отчет о результатах эксплуатации снятой с сопровождения базовой версии программного средства и ее архивации:

- причины и дата решения о прекращении сопровождения определенной базовой версии ПС и извещение пользователей;
- идентификатор ответственного лица, принявшего решение о прекращении сопровождения определенной версии ПС;
- дата и идентификатор лица, выполнившего архивацию определенной версии ПС;
- идентификаторы физических носителей информации архива, содержащих подлинники и дубликаты файлов и документов, снятой с сопровождения базовой версии ПС.

3.6.10. Отчет о результатах тиражирования базовых версий, конфигурациях и параметрах пользовательских версий программного средства:

- идентификаторы базовых версий ПС, поддерживаемых сопровождением и распространяемых пользователям;
- адреса архивов, содержащих физические носители файлов и документации каждой базовой версии ПС

	<p>распространяемой пользователям;</p> <ul style="list-style-type: none"> -краткая характеристика и адреса архивов, содержащих квалификационные тесты базовых версий ПС; -перечень идентификаторов пользователей, которым передана на эксплуатацию определенная базовая версия ПС; -идентификатор базовой версии, которая адаптировалась для эксплуатации каждым пользователем; -параметры среды пользователя, на которые адаптировалась определенная базовая версия ПС; -характеристики активности обращений пользователей к поставщику за консультациями и модификациями комплекса программ.
1.7	<p>3.7. Документы процессов эксплуатации программных средств</p> <p>3.7.1. Общее описание системы, в которой используется программное средство:</p> <ul style="list-style-type: none"> -назначение и идентификатор системы: <p><i>* вид деятельности, для информатизации которой предназначена система;</i></p>

* перечень объектов автоматизации и внешней среды, на которых используется система и ПС;

- описание системы:

* структура системы и назначение ее частей;

* сведения о программном продукте в целом и его частях, необходимые для корректной эксплуатации системы;

* описание функционирования системы и ее частей;

- описание взаимосвязей программного продукта с другими системами и ПС:

* перечень компонентов систем и ПС, с которыми связано данный программный продукт;

* описание регламента связей между системами и ПС;

* перечень функций, реализуемых каждой взаимодействующей системой и ПС;

- краткое описание ПС, перечень файлов, включая базу данных и файлы со справочной информацией для пользователей, описание аппаратуры и прочих ресурсов, необходимых для доступа и использования программного продукта в полном объеме;

- режимы работы программного продукта;

- терминалы, принтеры и другие входные и выходные устройства;

- необходимые процедуры, утилиты, в том числе процедуры для установки и инсталляции программного

продукта;

- форматы представления входной и выходной информации, их назначение, тип, объем;
- точность представления, скорость передачи, ожидаемое время реакции на операции пользователя;
- способ задания конца обработанной информации и другие требуемые соглашения;
- ограничения и наиболее типичные ошибки задания информации;
- описание используемой системы управления базой данных.

3.7.2.Описание административного управления программными средствами системы:

- концепции и обзоры системного управления программами и базами данных;
- документы, детализирующие концепцию процессов управления системой и ПС и требования к реализации каждой функции;
- информационная модель системы, комплекса программ, их атрибутов и операций;
- руководства для формализации и описания объектов управления системы и ПС;
- формализация непосредственной передачи управляющей информации между компонентами системы и ПС;
- документы, описывающие:

* передаваемые типы данных;

* формализованные объекты, их состояния, атрибуты, операции и извещения об обмене;

-классификатор объектов управления, отражающий взаимосвязь между классами объектов управления и правилами их применения;

-функции администратора программных средств:

* общие функции администрирования при применении данного ПС;

* процедуры по инсталляции и подготовке ПС к эксплуатации;

* контроль ввода заданий и выработки запроса на их выполнение;

* контроль представления результатов обработки заданий;

* способы и формы контроля исполнения заданий;

* динамическое управление процессом реализации заданий.

3.7.3. Руководство системного администратора программного средства:

- описание запуска системы управления и комплекса программ либо непосредственно с центрального компьютера, либо другим централизованным способом, либо через сеть;

- описание аппаратных и программных средств, требуемых для работы системы;

- технические характеристики используемых аппаратных устройств;
- структура, обзор назначения и функционирования каждого компонента комплекса программ;
- перечень входных команд, команд доступа к ПС и реакции на их выполнение;
- аварийные сообщения и другие выходные данные, формируемые для контроля комплекса программ;
- типовые времена выполнения основных функций ПС;
- последовательность действий для запуска системы и комплекса программ;
- перечень требуемых библиотек поддержки и интерфейсов системы;
- форма и средства регистрации дефектов и ошибок, возникающих в процессе эксплуатации ПС;
- перечень процедур, выполняемых системным администратором при установке ПС для конкретного выбранного окружения и конкретной конфигурации системы.

3.7.4.Общее описание руководства пользователей программного средства:

- порядок действий пользователя для установки и использования системы и ПС;
- краткое описание функций и характеристик ПС;
- описание внешней программной среды;

- перечень файлов, включая файлы базы данных, необходимых для применения ПС;
- порядок действий для продолжения или возобновления функционирования ПС в случаях возникновения непредвиденных ситуаций;
- организация и функционирование ПС с точки зрения пользователя;
- описание процедур, позволяющих фиксировать дефекты и ошибки;
- детальные, пошаговые действия пользователя при включении системы и дальнейшей работе с ней;
- ссылки на другие руководства системы и комплекса программ;
- перечень и пояснение выводимых системой сообщений.

3.7.5. Руководство оперативного пользователя программного средства:

- титульный лист, оформленный по правилам предприятия с учетом требований заказчика;
- ограничения на применение документа и указания на авторские права на программный продукт;
- введение:
 - * *область применения ПС;*
 - * *краткое описание функциональных возможностей;*

* требования к уровню подготовки пользователя;

* перечень эксплуатационной документации, с которыми необходимо предварительно ознакомиться пользователю;

-назначение и условия применения комплекса программ:

* теоретические основы данного комплекса программ, функции и решаемые задачи;

* виды деятельности и функции, для автоматизации которых предназначено данное программное средство;

* условия, при соблюдении (выполнении, наступлении), которых обеспечивается применение программного средства в соответствии с назначением, спецификациями требований и характеристиками системы;

* технические и административные операции для запуска решения функциональных задач;

* предостережения и предупреждения от ошибок пользователей;

* метод решения каждой задачи, их взаимодействие и ограничения;

-подготовка к работе:

* состав и содержание дистрибутивного носителя комплекса программ и данных;

* описание всех выполняемых функций, задач, процедур;

* описание операций технологического процесса обработки данных, необходимых для выполнения функций,

комплексов задач, процедур;

** порядок загрузки данных и программ;*

** порядок контроля и проверки работоспособности комплекса программ;*

- описание функциональных операций ПС - для каждой операции обработки данных должно быть указано:

** идентификатор и наименование операции;*

** условия, при соблюдении которых возможно выполнение операции;*

** подготовительные действия;*

** основные действия в требуемой последовательности функциональных операций;*

** исходные данные, необходимые для корректного функционирования комплекса программ;*

** информация для контроля корректного функционирования комплекса программ;*

** рекомендации как приостановить исполнение заданных функций и провести рестарт комплекса программ;*

** регистрация окончание исполнения заданной функции комплекса программ;*

** заключительные действия для завершения требуемой задачи;*

** оценка ресурсов, расходуемых на операцию или заданную функцию;*

-аварийные ситуации:

* *действия пользователя в случае несоблюдения условий выполнения технологического процесса, в том числе при отказах технических средств;*

* *действия пользователя по восстановлению программ и/или данных при отказе или обнаружении ошибок;*

* *действия в случаях обнаружения несанкционированного вмешательства в данные;*

-гарантии и обязательства по контракту на комплекс программ, а также условия отказа от них;

-рекомендации по обучению и освоению ПС, включая описание контрольного примера, правила его запуска и выполнения; приложения - детальные сведения о форматах исходных и результирующих данных, структуре файлов и экранов.

3.7.6. Инструкция по формированию и ведению информации базы данных:

-правила подготовки информации данных:

* *порядок отбора информации для включения в базу данных;*

* *правила подготовки и кодирования информации базы данных;*

* *формы ее представления и правила заполнения этих форм;*

* *порядок внесения изменений в информацию базы данных;*

-порядок и средства заполнения базы данных:

* *состав технических средств;*

* *правила, порядок, последовательность и описание процедур, используемых при заполнении базы данных, включая перенос данных на машинные носители информации;*

-процедуры изменения и контроля информации базы данных:

* *состав и последовательность выполнения процедур по контролю и изменению содержания базы данных;*

-порядок и средства восстановления информации базы данных:

* *описание средств защиты базы от разрушения и несанкционированного доступа;*

* *правила, средства и порядок проведения процедур по копированию и восстановлению базы данных.*

3.7.7. Паспорт на программное средство:

-общие сведения о программном средстве:

* *идентификатор и наименование ПС;*

* *его обозначение, присвоенное разработчиком;*

* *наименование предприятия-поставщика;*

-основные характеристики ПС:

- * *состав функций, реализуемых ПС;*
- * *описание принципов функционирования ПС;*
- * *общий регламент и режимы функционирования ПС;*
- * *сведения о возможности выбора и изменения режимов его работы;*
- * *сведения о совместимости ПС с другими системами и внешней средой;*

-комплектность:

- * *перечень эксплуатационных документов ПС;*
- * *все непосредственно входящие в состав ПС компоненты;*
- * *отдельные средства в комплексе программ, в том числе носители данных и эксплуатационные документы;*

-свидетельство (акт) о приемке:

- * *содержание и дата подписания акта о приемке ПС в промышленную эксплуатацию;*
- * *фамилии и должности лиц, подписавших акт о приемке;*

-гарантийные обязательства изготовителя (поставщика):

- * сроки и ограничения гарантии на комплекс программ в целом;
- * сроки и ограничения гарантии отдельных составных частей, если они не совпадают с условиями гарантии ПС в целом;
- сведения о текущем состоянии комплекса программ:
- * сведения о выявленных дефектах;
- * замечания по эксплуатации и аварийным ситуациям, принятые меры;
- * сведения о изменениях в программном средстве с указанием основания, даты и содержания изменения;
- сведения о выполнении регламентных, профилактических работ и их результатах.

3.7.8. Пользовательская документация на коммерческие пакеты – закрытые коробки программных средств по стандарту ISO 9127:

-документация внутри коробки:

- * цель и назначение ПС;
- * справочная документация:
- * идентификация пакета ПС;
- * составные части пакета ПС;

- * функциональное описание ПС;
- * инсталляция (сборка) ПС;
- * использование ПС;
- * технологическая информация по ПС;
- * тестирование при применении;
- * информация по контрактам;
- * глоссарий;
- * индекс;
- * замечания для конечного пользователя;
- обучающая документация;
- документация для быстрых справок;
- документация на внешней упаковке пакета:
- * цель ПС;
- * содержание пакета:

- * идентификация пакета;
- * цель и область применения;
- * окружение;
- * вход;
- * выход;
- * ограничения на данные в файлах;
- * инструкция для пользователя;
- * дополнительная информация;
- * информация по контрактам;
- * адреса сервиса для потребителя;
- * предметное обеспечение (спецификация);
- * стандарты и законы;
- * независимая сертификация;
- * код продукта;

-цена.

3.7.9. Руководство по подготовке документации и обучению специалистов применению программного средства:

- виды и уровни обучения и категории персонала, требующие обучения применению программного средства;
- требования к начальным квалификации и опыту, необходимым операторам, администраторам и техническому персоналу;
- документация планов и требований по обучению специалистов;
- руководства для обучения, включая материалы и средства автоматизации, используемые для обучения разных категорий специалистов;
- план регулярного обучения персонала, контроля и учета результатов обучения с методами оценки достигнутой квалификации специалистов;
- требования и содержание сертификата специалиста, гарантирующего, что соответствующие категории обученного персонала готовы для выполнения запланированных действий и решения задач с применением определенного программного средства.

2.1	СЕРТИФИКАЦИОННЫЕ ИСПЫТАНИЯ ПРОГРАММНОГО ПРОДУКТА НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ
-----	---

Порядок сертификационных испытаний сложного программного продукта

Сертификационные испытания программного продукта на соответствие требованиям

Порядок сертификационных испытаний сложного программного продукта:

- требования технического задания и спецификаций требований на программный продукт и систему;
- Программа испытаний программного продукта на соответствие утвержденным требованиям;
- методики и протоколы испытаний по разделам Программы и требований спецификаций на программный продукт;
- функциональные испытания для оценивания характеристик и качества программного продукта;
- комплект адекватной эксплуатационной и технологической документации на программный продукт.

Испытания надежности функционирования программного продукта:

- экспериментальные методы оценивания надежности программного продукта в штатном режиме;
- форсированные испытания для оценивания надежности программного продукта;
- повышение надежности комплекса программ путем оперативного контроля и рестарта.

Испытания функциональной безопасности программного продукта:

- оценивание эффективности испытаний функциональной безопасности программного продукта;
- учет ограниченности ресурсов для обеспечения испытаний функциональной безопасности программного продукта;
- удостоверение достигнутой функциональной безопасности системы с программным продуктом.

Испытания производительности и динамического использования ресурсов ЭВМ программным продуктом:

- оценивание нарушения временного баланса между длительностью решения задач и производительностью ЭВМ;
- оценивание предельной пропускной способности системы с конкретным программным продуктом;
- определение качества динамического функционирования программного продукта при перегрузках.

Процесс сертификации программных продуктов включает:

- анализ и выбор разработчиком или заказчиком (заявителем), компетентных в данной области органа и аттестованной лаборатории для выполнения сертификационных испытаний;
- подачу заявителем заявки на испытания в орган сертификации и принятие сертификаторами решения по заявке, выбор схемы сертификации, заключение договора на сертификацию;
- идентификацию требований к версии программного продукта, подлежащих испытаниям;
- выполнение сертификационных испытаний версии программного продукта сертификационной лабораторией;
- анализ полученных результатов и принятие решения лабораторией и/или органом сертификации о возможности выдачи заявителю сертификата соответствия;
- выдачу органом сертификации заявителю – сертификата и лицензии на применение знака соответствия и на выпуск сертифицированной продукции – версий программного продукта;
- осуществление инспекционного контроля органом сертификации сертифицированной продукции;
- проведение заявителем корректирующих мероприятий при нарушении соответствия продукции установленным требованиям и при неправильном применении знака соответствия.

Программа и методики испытаний комплексов программ на соответствие требованиям

Сертификационные испытания программного продукта на соответствие требованиям

Порядок сертификационных испытаний сложного программного продукта:

- требования технического задания и спецификаций требований на программный продукт и систему;
- Программа испытаний программного продукта на соответствие утвержденным требованиям;
- методики и протоколы испытаний по разделам Программы и требований спецификаций на программный продукт;
- функциональные испытания для оценивания характеристик и качества программного продукта;
- комплект адекватной эксплуатационной и технологической документации на программный продукт.

Испытания надежности функционирования программного продукта:

- экспериментальные методы оценивания надежности программного продукта в штатном режиме;
- форсированные испытания для оценивания надежности программного продукта;
- повышение надежности комплекса программ путем оперативного контроля и рестарта.

Испытания функциональной безопасности программного продукта:

- оценивание эффективности испытаний функциональной безопасности программного продукта;
- учет ограниченности ресурсов для обеспечения испытаний функциональной безопасности программного продукта;
- удостоверение достигнутой функциональной безопасности системы с программным продуктом.

Испытания производительности и динамического использования ресурсов ЭВМ программным продуктом:

- оценивание нарушения временного баланса между длительностью решения задач и производительностью ЭВМ;
- оценивание предельной пропускной способности системы с конкретным программным продуктом;
- определение качества динамического функционирования программного продукта при перегрузках.

Программа испытаний должна содержать следующие четко сформулированные разделы:

- объект испытаний, его назначение и перечень основных документов, определивших его разработку;
- цель испытаний, с указанием всех требований технического задания, характеристик качества, подлежащих проверке, и ограничений на проведение испытаний;
- собственно Программу испытаний, содержащую проверку комплектности спроектированного программного продукта в соответствие с техническим заданием, план и график проведения тестирования для проверки по всем разделам требований технического задания и дополнительных требований, formalizованных отдельными решениями разработчиков и заказчика;
- перечень методик испытаний, однозначно определяющие все требования, понятия проверяемых функций и характеристики качества, условия и сценарии тестирования, инструментальные средства, используемые для испытаний;
- перечень методик обработки и оценки результатов тестирования по каждому разделу Программы испытаний.

2.2

Испытания эксплуатационной документации на соответствие требованиям к программному продукту

Ориентировочный комплект основных документов при сертификации программного продукта состоит из трех групп:

- **базовые нормативные документы** характеристик качества программных продуктов и их производства, а также подготовленные разработчиками на их основе Программа, Руководство и инструкции, предъявляемые испытателями сертифицирующей организации для испытаний версии программного продукта:

Базовые нормативные документы характеристик качества программного продукта и их производства:

- концепция, требования и Руководство по улучшению деятельности – Системы менеджмента качества – **ISO 9000:2000** или версия модели зрелости **CMMI**;
- административное управление системой качества предприятия, разрабатывающего программный продукт – ISO 900003;
- жизненный цикл программных средств – **ISO 12207, ISO 15504**;
- сопровождение и конфигурационное управление версиями программного продукта **ISO 14764, ISO 15846**;
- характеристики качества программных продуктов – **ISO 9126, ISO 14598, ISO 25000**;
- функциональной безопасности программных продуктов – **IEC 61508, ISO 15408**;

- документирования программных продуктов – **ISO 15910, ISO 18019**;
 - комплект должностных инструкций, определяющих ответственность, полномочия и порядок взаимодействия всего персонала, для производства конкретного программного продукта.
-
- **исходные документы**, отражающие результаты предварительных испытаний и характеристики сертифицируемого программного продукта, подготавливаемые руководством производства продукта (заявителем) для сертификации его качества:

Исходные документы, отражающие результаты предварительных испытаний и характеристики сертифицируемого программного продукта:

- Техническое задание – требования к функциям, характеристикам, качеству программного продукта, системы и внешней среды;
- Договор заказчика с производителем программного продукта;
- Писание целей, требований и обязательств производителя в области критериев качества процессов производства и программного продукта;
- Полные тексты программ, содержание базы данных и технологической документации версии программного продукта;

- комплект эксплуатационных документов, поставляемых заказчику и пользователям для применения версии программного продукта;
- план, Программа и методики испытаний, применения и оценки качества программного продукта;
- методики сопровождения, анализа и утверждения версий программного продукта;
- методика конфигурационного управления, утверждения, хранения, защиты, копирования версий программного продукта и сопровождающих документов в архиве предприятия;
- технические условия на версию программного продукта, базу данных управления конфигурацией и эксплуатационную документацию для тиражирования и серийного производства;
- руководство по генерации и инсталляции пользовательских версий и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;
- акт о завершении предварительных испытаний и готовности к поставке и/или предъявлению для сертификационных испытаний версии программного продукта.

• **результатирующие документы** сертификационных испытаний версии программного продукта, представляемые органу сертификации, заявителю и руководству производства для утверждения сертификата и знака качества

Результирующие документы сертификационных испытаний программного продукта:

- Договор заявителя с сертифицирующей организацией на проведение испытаний версии программного продукта;
- отчет заявителя о наличии, актуальности и систематичности тестирования и оформления программного продукта и документации на протяжении жизненного цикла программного продукта;
- отчет сертификаторов о рабочем состоянии Программы и методик проведения сертификационных испытаний в соответствии с требованиями Договора на сертификацию с заявителем;
- результаты аттестации имитаторов внешней среды и генераторов динамических тестов для сертификационных испытаний версии программного продукта;
- результаты выполнения планов и методик сертификационных испытаний, протоколы соответствия испытаний предъявляемым требованиям, утвержденные сертификаторами и согласованные с заявителями;
- протоколы достигнутых при сертификационных испытаниях характеристиках качества версии программного продукта;
- акт результатов испытаний реальных характеристик версии программного продукта, выводы о их соответствии требованиям к характеристикам программного продукта;
- сертификат версии программного продукта и обеспечения его жизненного цикла, лицензия на применение знаков соответствия.

3.1	<p>Оценка процесса разработки программного обеспечения</p> <p>Проект стандарта ИСО/МЭК 15504 состоит из следующих частей, под общим названием <i>Оценка процесса разработки ПО</i></p> <p>Часть 1: <i>Понятия и вводное руководство</i> (информационная)</p> <p>Часть 2: <i>Эталонная модель процессов и возможности процесса</i> (нормативная)</p> <p>Часть 3: <i>Выполнение оценки</i> (нормативная)</p> <p>Часть 4: <i>Руководство по выполнению оценки</i> (информационная)</p> <p>Часть 5: <i>Модель оценки и показательное руководство</i> (информационная)</p> <p>Часть 6: <i>Руководство по компетенции эксперта-консультанта</i> (информационная)</p> <p>Часть 7: <i>Руководство для использования в улучшение процесса</i> (информационная)</p> <p>Часть 8: <i>Руководство для использования в определении возможности процесса поставщика</i> (информационная)</p> <p>Часть 9: <i>Словарь</i> (информационная)</p>

Основные положения

ИСО/МЭК 15504 обеспечивает руководство для оценки процесса разработки программного обеспечения. Оно может использоваться организациями для планирования, менеджмента, текущего контроля, управления и совершенствования приобретения, поставки, разработки, функционирования, развития и поддержки программного обеспечения.

ИСО/МЭК 15504 обеспечивает структурированный подход к оценке процесса разработки ПО для следующих целей:

- организацией или от ее имени с целью понимания состояния собственных процессов для совершенствования (улучшения) процесса разработки ПО*
- организацией или от ее имени с целью определения пригодности собственных процессов для удовлетворения специфического требования или класса требований*
- организацией или от ее имени с целью определения пригодности процессов другой организации для специфического контракта или класса контрактов.*

Руководство оценки поощряет самостоятельную оценку, определяет управление для оцениваемых процессов, принимает во внимание контекст, в котором оцениваемые процессы функционируют, вырабатывает набор рейтингов процесса (профиля процесса), соответствует всем предметным областям и размерам организаций.

Использование оценки процесса разработки ПО внутри организации должно утвердить:

- культуру постоянного улучшения и учреждения соответствующих механизмов для поддержки и сопровождения этой культуры;*
- инжиниринг процессов, с целью удовлетворения деловых требований;*
- оптимальное использование ресурсов.*

Пользователи извлекут выгоду из использования оценки, определенной в данном стандарте. Ее использование в определения возможности процесса позволит:

- уменьшить неопределенность и риск в выборе поставщика программных систем при заключении контракта;*
- поместить соответствующие средства управления в наиболее рискованные места жизненного цикла проекта;*
- обеспечить определенную количественную основу для выбора финансовых потребностей, требований и оценить стоимость проекта относительно возможностей конкурирующих поставщиков.*

Основные преимущества стандартизированного подхода к оценке процесса разработки заключаются в том, что:

- общественности представлена общедоступная модель;*
- достигнуто общее понимание в использовании оценки для улучшения процесса и определения его возможностей;*
- облегчена процедура определения возможности поставки оборудования;*

- процесс разработки ПО управляется и регулярно просматривается в свете опыта использования;*
- подход может быть изменен только с международного согласия;*
- способствует гармонизации существующих моделей и схем оценки.*

3.2

Выполнение оценки

Подраздел определяет набор минимальных требований для выполнения оценки. Они увеличивают вероятность того, что результаты оценки процессов являются целевыми, беспристрастными, непротиворечивыми, повторимыми и представительными, и определяют обстоятельства, при которых результаты оценки являются сравнимыми.

Требования формируют часть структуры оценки процесса, который:

- поддерживает самооценку;*
- принимает во внимание контекст, в котором оцененные процессы функционируют;*
- производит наборы рейтингов процесса (профили процесса);*
- через атрибуты процесса, применимые ко всем процессам, направляют возможность процессов, чтобы достигнуть их целей;*

- является соответствующим всем предметным областям и размерам организации.

Чтобы увеличить непротиворечивость оценок для оцениваемых процессов, необходимо обеспечить требования для проведения оценок. Требования помогают гарантировать, что выход оценки непротиворечив, обеспечивает подтверждение, чтобы доказать обоснованность оценки и проверять согласие с требованиями.

Вход должен быть определен до оценки и одобрен заказчиком оценки. Вход оценки, как минимум, должен определять следующее.

1. Личность заказчика оценки и его отношение к оцениваемой организации.

2. Цель оценки.

3. Сферу оценки, включая:

- а) процессы, которые будут исследованы внутри организации;
- б) самый верхний уровень возможности, который должен исследоваться;
- в) часть организации, которая разворачивает эти процессы;
- г) контекст процесса, который, как минимум, включает: размер организации; демографию организации; предметную область продукта или услуг организации; размер, уязвимость и сложность изделий или услуг; качественные характеристики изделий (см., например, ИСО/МЭК 9126-91 Характеристики качества ПО).

4. Ограничения целостности оценки, которые могут включать:

- а) доступность ключевых ресурсов;
- б) максимальное количество времени, которое должно использоваться для оценки;
- в) специфические процессы, которые должны быть исключены из оценки;
- г) минимальный, максимальный или специфический типовой размер или покрытие, которое желательно для оценки;
- д) собственность на выполнение оценки и любые ограничения на их использование;
- е) средства управления на информацию, следующей из соглашения конфиденциальности.

5. Подлинность модели, используемой в пределах оценки, которая должна быть совместима с хорошей модель инжиниринга ПО, и встречающая требования, рассмотренные в подразделе 4.2.

6. Личности экспертов-консультантов, включая компетентного эксперта-консультанта, ответственного за оценку.

7. Личности экспертов-консультантов и персонала поддержки со специфическими обязанностями для оценки.

8. Любая дополнительная информация, которая будет собрана в течение оценки, чтобы поддерживать улучшение процесса или определение возможности процесса.

Любые изменения во входах оценки должны быть согласованными с заказчиком и зарегистрироваться в записи оценки. Заказчик оценки должен проверять, что эксперт-консультант, который должен брать

ответственность за оценку и наблюдать за ней (компетентный эксперт-консультант), имеет необходимую компетентность и умения. Эксперт-консультант должен подтвердить требования заказчика, чтобы приступить к оценке.

Компетентный эксперт-консультант должен гарантировать, что оценка проводится в соответствии с определенными требованиями и принимает во внимание релевантные руководства в других частях ИСО/МЭК 15504.

Эксперт-консультант должен гарантировать, что все другие участники оценки проинформированы о целях, сфере и методе оценки. По завершении оценки, компетентный эксперт-консультант должен подтвердить, что требования были выполнены.

Оценка должна проводиться согласно зарегистрированному процессу, который является способным удовлетворить цель оценки.

Оценка процесса, в минимуме, должен содержать следующие действия.

1. Планирование. План оценки должен быть разработан и зарегистрирован, определяя, в минимуме: требуемые входы; действия, которые нужно выполнить для проведения оценки; ресурсы и план, назначенный этим действиям; выбор и определенные обязанности экспертов-консультантов и участников организации оценки; критерии для проверки исполнения требований.

2. Сбор данных. Данные, требуемые для оценки процессов в пределах области оценки, должны быть собраны систематическим и упорядоченным способом, применяя, в минимуме, следующие принципы:

- должны быть явно идентифицированы стратегия и методы сбора и анализа данных;*

- должно быть установлено соответствие между процессами организации, определенные в сфере оценки через совместимую модель, используемую для оценки, и процессами, определенными в эталонной модели;
- каждый процесс, идентифицированный в области оценки, должен быть оценен на основе целевого доказательства;
- целевое доказательство, собранное для каждого атрибута оцененного процесса, должно быть достаточным для удовлетворения цели оценки и ее сферы;
- цель - утвержденное доказательство, основанное на показателях оценок атрибута процесса, которые подтверждают решения эксперта-консультанта, должна быть зарегистрирована, чтобы обеспечить основу для проверки оценок.

Эти записи должны включать:

- а) методы и исследованные рабочие продукты;
- б) имена личностей, обеспечивающих информацию;
- в) обсуждение оценок;

3. Проверка правильности данных. Собранные данные должны быть утверждены, гарантируя, что утвержденные данные достаточно покрывают область оценки.

4. Ранжирование процесса. Оценка должна быть назначена и утверждаться для каждого атрибута процесса, до, и включая, самый высокий уровень возможности, определенный в области оценки, для определенной части организации.

Набор рейтингов атрибута процесса должен быть зарегистрирован как профиль процесса для определенной части организации.

Чтобы обеспечивать основу для повторяемости оценки, определенный набор показателей оценки в совместимой модели должен использоваться в течение оценки, чтобы поддерживать суждения эксперта-консультанта в рейтинге атрибутов процесса.

5. Результаты. Результаты оценки, включая, как минимум, выходы, определенные ниже (запись оценки), должны быть зарегистрированы и сообщаться заказчику оценки.

Критерии для квалификации компетентного эксперта-консультанта должны быть зарегистрированы.

Эксперты-консультанты, участвующие в оценке, должны иметь доступ к соответствующим материалам, описывающим как выполнить оценки, и необходимую компетентность, использовать любые приборы или инструментальные средства, выбранные, чтобы поддерживать оценку.

Запись оценки, как минимум, должна содержать:

- дату оценки;*
- вход оценки (исходные данные);*
- используемый метод оценки и инструментальные средства;*
- имена экспертов-консультантов, которые провели оценку, включая компетентного эксперта-консультанта, ответственного за оценку;*

	<ul style="list-style-type: none"> <input type="checkbox"/> набор профилей процесса, следующих из рейтингов (т. е. один профиль для каждого оцененного процесса), <input type="checkbox"/> расположение записей объективного доказательства, обеспечивающего оценки атрибута процесса в профилях процесса, и эксперт-консультант (-ов), ответственных за решение оценки, <input type="checkbox"/> любая дополнительная информация, собранная в течение оценки, которая была идентифицирована в начале оценки, чтобы поддерживать улучшение процесса или определение возможности процесса, <input type="checkbox"/> время, потраченное экспертом-консультантом, и уровень возможности процесса или уровня, как определено в эталонной модели.
3.3	<p>Запись должна содержать следующее утверждение: эта оценка провелась в соответствии с требованиями ISO 15504. Оценка процесса разработки программного обеспечения.</p> <p>СЕРТИФИКАЦИЯ ПРОЦЕССОВ ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ И СИСТЕМ КАЧЕСТВА ПРЕДПРИЯТИЙ</p> <p>Подготовка и документирование организации процессов сертификации производства и системы качества предприятия</p> <ul style="list-style-type: none"> <input type="checkbox"/> - общие требования нормативных документов на продукцию сертифицируемого производства; <input type="checkbox"/> - положение о подразделениях и должностные инструкции, обязанности и полномочия специалистов, связанных с реализацией процессов производства; <input type="checkbox"/> - особенности и состав документации процессов и результатов сертификации – аудита технологии и системы качества производства программных средств:

- Заявка на проведение сертификации производства программных средств;
- Регистрационная карта сертифицируемого производства программных средств;
- Заключение по результатам рассмотрения заявки на сертификацию;
- Задание на проведение сертификационных испытаний;
- План сертификации производства программного средства;
- Заключение по результатам сертификационных испытаний производства программного средства;
- Документы сертификационной лаборатории – Программа испытаний и методики аудита – испытаний производства должны содержать:
 - Определение объекта сертификационных испытаний;
 - Цель испытаний– аудита;
 - Требования к Программе сертификационных испытаний;
 - Требования к документации испытаний;
 - Средства автоматизации и порядок испытаний;
 - Методики испытаний по каждому разделу требований производства;

- Формы отчетности выполнения и результатов испытаний;
- Отчет о проверках организации сертификационных испытаний аудита производства и системы качества программного продукта;
- Передача главным аудитором отчета о проверках и организации сертификационных испытаний производства и системы качества программного продукта клиенту – заявителю.

Организация сертификационных испытаний производства и систем менеджмента качества программных средств:

- условия и документы для обеспечения процессов сертификационных испытаний программных средств;
- программа и методики испытаний – аудита производства и системы менеджмента качества предприятия;
- организация работ по сертификации производства и системы менеджмента качества программных средств;
- первоначальное заседание комиссии по сертификации;
- заочная оценка качества производства программного средства;
- подготовка к аудиту – испытаниям на месте производства программных средств;

- одобрение Программы и методик испытаний и передача ее аудиторам;
- проведение аудита – испытаний на месте производства и подготовка акта по результатам аудита предприятия и качества производства программных средств;
- источники информации о состоянии производства;
- протоколы и отчет под руководством главного аудитора о результатах испытаний качества производства;
- критерии оценки качества производства на соответствие требованиям комиссии аудиторов;
- результаты аудита, выводы и рекомендации комиссии, оформленные в виде акта;
- расширение или сужение области сертификации производства;
- завершение сертификации, выдача и регистрация сертификата по результатам аудита – испытаний производства программных средств;
- инспекционный контроль состояния сертифицированного производства программных средств.

3.4

ЕДИНАЯ СИСТЕМА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

ГОСТ 19.001-77 ОБЩИЕ ПОЛОЖЕНИЯ

ГОСТ 19.101-77 ВИДЫ ПРОГРАММ И ПРОГРАММНЫХ ДОКУМЕНТОВ

ГОСТ 19.103-77 ОБОЗНАЧЕНИЕ ПРОГРАММ И ПРОГРАММНЫХ ДОКУМЕНТОВ

ГОСТ 19.104-78 ОСНОВНЫЕ НАДПИСИ

ГОСТ 19.105-78 ОБЩИЕ ТРЕБОВАНИЯ К ПРОГРАММНЫМ ДОКУМЕНТАМ

ГОСТ 19.106-78 ТРЕБОВАНИЯ К ПРОГРАММНЫМ ДОКУМЕНТАМ, ВЫПОЛНЕННЫМ ПЕЧАТНЫМ СПОСОБОМ

ГОСТ 19.201-78 ТЕХНИЧЕСКОЕ ЗАДАНИЕ. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ

ГОСТ 19.202-78 СПЕЦИФИКАЦИЯ. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ

ГОСТ 19.401-78 ТЕКСТ ПРОГРАММЫ. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ

ГОСТ 19.402-78 ОПИСАНИЕ ПРОГРАММЫ

ГОСТ 19.404-79 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ

**ГОСТ 19.503-79 РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА.
ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ**

**ГОСТ 19.504-79 РУКОВОДСТВО ПРОГРАММИСТА. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И
ОФОРМЛЕНИЮ**

ГОСТ 19.505-79 РУКОВОДСТВО ОПЕРАТОРА. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ

ГОСТ 19.506-79 ОПИСАНИЕ ЯЗЫКА. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ

3.5

СИСТЕМА ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ НА АСУ

**ГОСТ 24.202-80 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТА «ТЕХНИКО-ЭКОНОМИЧЕСКОЕ
ОБОСНОВАНИЕ СОЗДАНИЯ АСУ»**

ГОСТ 24.203-80 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ОБЩЕСИСТЕМНЫХ ДОКУМЕНТОВ

ГОСТ 24.204-80 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТА «ОПИСАНИЕ ПОСТАНОВКИ ЗАДАЧИ»

ГОСТ 24.205-80 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТОВ ПО ИНФОРМАЦИОННОМУ ОБЕСПЕЧЕНИЮ

ГОСТ 24.207-80 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТОВ ПО ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

ГОСТ 24.208-80 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТОВ СТАДИИ «ВВОД В ЭКСПЛУАТАЦИЮ»

ГОСТ 24.209-80 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТОВ ПО ОРГАНИЗАЦИОННОМУ ОБЕСПЕЧЕНИЮ

ГОСТ 24.210-82 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТОВ ПО ФУНКЦИОНАЛЬНОЙ ЧАСТИ

ГОСТ 24.211-82 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТА «ОПИСАНИЕ АЛГОРИТМА»

ГОСТ 24.301-80 ОБЩИЕ ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ ТЕКСТОВЫХ ДОКУМЕНТОВ

ГОСТ 24.302-80 ОБЩИЕ ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ СХЕМ

ГОСТ 24.304-82 ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ ЧЕРТЕЖЕЙ

Перечень показателей текущего контроля

Номер показателя	Значение показателя
1.1.2.1	Способность формировать документы предварительных требований, спецификаций и ресурсов для разработки программного средства
1.1.2.2	Способность формировать документы процессов проектирования и выбора характеристик качества программного средства
1.1.2.3	Способность формировать документы процессов разработки и программирования компонентов программных средств
1.1.2.4	Способность формировать документы верификации и тестирования компонентов программных средств
1.1.2.5	Способность формировать документы квалификационного тестирования, испытаний и оценивания качества программных средств
1.1.2.6	Способность формировать документы сопровождения и конфигурационного управления версиями программного средства
1.1.2.7	Способность формировать документы процессов эксплуатации программных средств
1.1.11.1	Соответствие документов ГОСТ
2.1.2.1	Понимание процесса разработки программного обеспечения
2.1.5.1	Понимание процесса верификации и аттестации программного обеспечения
2.1.9.1	Понимание основ метрологии программных продуктов
2.1.10.1	Понимание порядка сертификационных испытаний программного продукта
2.1.10.2	Понимание программы и методики испытаний комплексов программ на соответствие требованиям
2.1.10.3	Способность формировать программу и применять методики испытаний программ
2.1.11.1	Способность использовать базовые нормативные документы характеристики качества программного продукта

	и их использования
2.1.11.2	Способность формировать документы, отражающие результаты предварительных испытаний и характеристики сертифицируемого программного продукта
2.1.11.3	Способность формировать документы сертификационных испытаний программного продукта
3.1.2.1	Понимание процесса разработки программного обеспечения
3.1.5.1	Понимание стандартов оценки процессов разработки программного обеспечения
3.1.9.1	Понимание процедуры выполнения оценки программного обеспечения
3.1.10.1	Понимание процесса подготовки и документирования организации процессов сертификации производства и системы качества предприятия
3.1.10.2	Понимание процесса организации сертификационных испытаний производства и систем качества программных средств
3.1.11.1	Способность применять ГОСТ ЕДИНАЯ СИСТЕМА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ
3.1.11.2	Способность применять ГОСТ СИСТЕМА ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ НА АСУ

3. РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ПРАКТИКИ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ НА ТЕКУЩЕМ КОНТРОЛЕ

Учебная практика на текущем контроле не предусмотрена.

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

МДК.03.01 Технология разработки программного обеспечения, семестр 7, Дифференцированный зачет

Автоматический контроль по результатам текущего контроля
Текущий контроль №1

Текущий контроль №2
Текущий контроль №3
Текущий контроль №4
Текущий контроль №5

МДК.03.02 Инструментальные средства разработки программного обеспечения, семестр 7, Дифференцированный зачет

Результаты обучения (освоенные умения, усвоенные знания) (Из стандарта)	Оцениваемые дидактические единицы	Основные показатели оценки результата	№ задания относящийся к показателю оценивания	Формы и методы контроля и оценки		Индекс занятия текущего контроля
				Методы:	Формы	
ПК.3.1	1.2	1.37539.1	1	Опрос	Индивидуальные задания	1.1.1, 1.1.2, 1.1.3
ПК.3.1	1.4	1.37541.1	1	Опрос	Индивидуальные задания	2.1.1, 2.1.2, 2.1.3, 2.1.4, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.5, 2.2.6, 2.2.7, 2.2.8, 2.2.9, 2.2.10, 2.2.11, 2.2.12,

ПК.3.2						2.2.13, 2.2.14, 2.2.15, 2.2.16, 2.2.17, 2.2.18, 2.2.19, 2.2.20, 2.2.21
ПК.3.3						
ПК.3.4						
ПК.3.6						
ПК.3.1	1.7	1.37544.1	1	Опрос	Индивидуальн ые задания	2.2.5, 2.2.15, 2.2.21
ПК.3.3						
ПК.3.4						
ПК.3.6						
ПК.3.2	1.8	1.37545.1	1	Опрос	Индивидуальн ые задания	3.1.1, 3.1.2
ПК.3.3						
ПК.3.5						
ПК.3.6						
ПК.3.1	2.2	2.37550.1	1	Опрос	Индивидуальн ые задания	2.2.5, 2.2.15, 2.2.21
ПК.3.3						
ПК.3.4						
ПК.3.6						

Перечень заданий

Номер задания	Задания
1	Варианты индивидуальных заданий

1. Разработать программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.
2. Разработать программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.
3. Разработать приложение Windows «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.
4. Разработать приложение Windows «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и желательно (но не обязательно) несколько математических функций.

5. Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.
6. Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.
7. Разработать программный модуль «Автосервис». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.
8. Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.
9. Разработать программный модуль «Картотека агентства недвижимости», предназначенный для использования работниками агентства. В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки на обмен (куплю, продажу) производится поиск подходящего

варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.

10. Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.
11. Разработать программный модуль «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты. В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиабилетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.
12. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.
13. Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.

14. Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

Перечень показателей

Номер задания	Задания
1.2.1	Умение анализировать задание и определять функционал
1.4.1	Умение проектировать программные продукты с заданной функциональностью
1.7.1	Уметь применять CASE-средства при проектировании программных продуктов
1.8.1	Умение работать в коллективе
2.2.1	Умение получать код программы с заданной функциональностью

МДК.03.03 Документирование и сертификация, семестр 7, Дифференцированный зачет

Автоматический контроль по результатам текущего контроля
Текущий контроль №1
Текущий контроль №2
Текущий контроль №3