

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Кузбасский государственный технический университет
имени Т.Ф. Горбачева»

Кафедра информационных и автоматизированных
производственных систем

Составитель
О. Н. Ванеев

МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Методические указания к курсовому проекту для студентов
направления подготовки 09.03.02 «Информационные системы
и технологии» очной формы обучения

Рекомендовано учебно-методической комиссией направления
09.03.02 «Информационные системы и технологии» в качестве
электронного издания для использования в учебном процессе

Кемерово 2017

Рецензенты:

Трусов А. Н. – доцент кафедры информационных и автоматизированных производственных систем

Чичерин И. В. – председатель учебно-методической комиссии направления 09.03.02. «Информационные системы и технологии»

Ванеев Олег Николаевич

Методы и средства проектирования информационных систем и технологий: методические указания к курсовому проекту [Электронный ресурс]: для студентов направления подготовки 09.03.02 «Информационные системы и технологии» очной формы обучения / сост. О. Н. Ванеев; КузГТУ. – Кемерово, 2017. – Электрон. дан. – Систем. требования: Pentium IV; ОЗУ 256 Мб; Windows XP; мышь. – Загл. с экрана.

В данных методических указаниях изложены содержание, порядок выполнения курсового проекта.

© КузГТУ, 2017

© Ванеев О. Н.,
составление, 2017

1. ОБЩИЕ ПОЛОЖЕНИЯ

Курсовой проект является аттестационной работой по курсу «Методы и средства проектирования информационных систем и технологий». При выполнении проекта студент должен показать качество освоения материала курса и практические навыки:

- по применению методик проектирования информационных систем (ИС);
- по формированию документов, сопутствующих этапам проектирования информационных систем;
- по использованию CASE средств проектирования ИС;
- по выполнению различных этапов проектирования:
 - анализ предметной области;
 - формирование требований;
 - проектирование ИС;
 - разработка модели хранения данных;
- по применению различных технологий программирования при реализации программного кода.

2. ТЕМА КУРСОВОГО ПРОЕКТА

Темой курсового проекта является разработка информационного сервиса автоматизации бизнес-процесса предприятия. Информационный сервис должен предусматривать использование хранилища данных, развитую клиентскую часть (не менее 4-5 форм), реализацию достаточно сложной логики прикладной области.

Задание на курсовую работу определяется на основе данных преддипломной практики, однако тема может быть изменена руководителем. Задание содержание должно включать проектирование модуля информационной системы предприятия, автоматизирующей определенный бизнес-процесс.

Курсовой проект должен содержать разделы в соответствии с пунктами разд. 4 данных методических указаний.

3. ФОРМА ПРЕДСТАВЛЕНИЯ ПРОЕКТНЫХ РЕШЕНИЙ И ИСПОЛЬЗУЕМЫЕ ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА

Разработку курсового проекта необходимо вести на основе объектно-ориентированного подхода с представлением проектных решений в виде совокупностей моделей, отображенных диаграммами UML. Диаграммы необходимо строить в среде одного из CASE средств Rational Rose, Enterprise Architect, Microsoft Visio или других.

Модели, отображающие результаты разработки, необходимо группировать в пакеты или «области просмотра» (*View*). В соответствии с традиционно выделяемыми процессами разработки ИС необходимо задать следующие области просмотра или пакеты верхнего уровня (рис. 1).

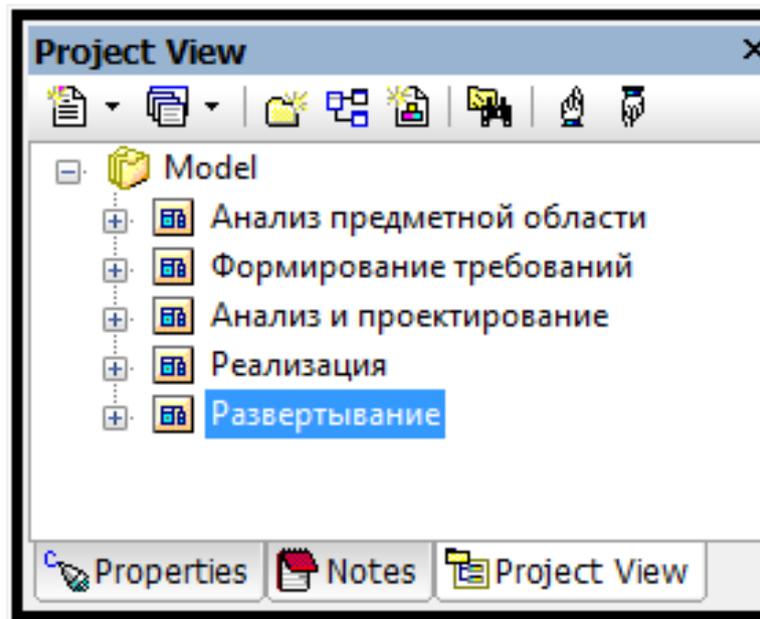


Рис. 1. Пакеты верхнего уровня в модели разработки информационной системы

Каждая диаграмма должна быть пояснена соответствующим описанием введенных в ней элементов и сценариев.

4. СОДЕРЖАНИЕ КУРСОВОГО ПРОЕКТА

4.1. Разработка технического задания

4.1.1. Исходная постановка задачи проектирования

В качестве исходной постановки обычно определяются некоторые предварительные функции разрабатываемой системы и ее примерное назначение.

Например, «система, формирующая отчеты по заказам клиентов».

Для конкретизации данной постановки, определения требований системы необходимо знать, какие конкретно процессы, процедуры деятельности предприятия будет автоматизировать разрабатываемая система, какой внешний вид должна иметь система. Данная работа выполняется на основе анализа условий предприятия.

4.1.2. Анализ условий предприятия

На данном этапе необходимо проанализировать деятельность, выполняемую предприятием, и предварительно наметить место предполагаемой системы в деятельности предприятия.

Для выполнения этой работы необходимо выявить:

- цели функционирования предприятия, то есть цели его бизнес-процессов;
- организационную структуру предприятия;
- состав его бизнес-процессов (БП);
- состав автоматизируемых бизнес-процессов и БП, связанных с ними;
- содержание бизнес-процессов;
- определить состав бизнес-объектов;
- определить типы данных, используемые в предприятии;
- выявить бизнес-решения, требующие автоматизации.

Решение данных задач необходимо отобразить в виде соответствующих моделей: «цели бизнес-процессов», «организационная структура», «состав бизнес-процессов», «содержание бизнес-процессов», «состав классов бизнес-объектов», «бизнес-решения, требующие автоматизации». Диаграммы, отображающие данные модели, необходимо разместить в соответствующие пакеты области просмотра «анализ предметной области» (рис. 2).

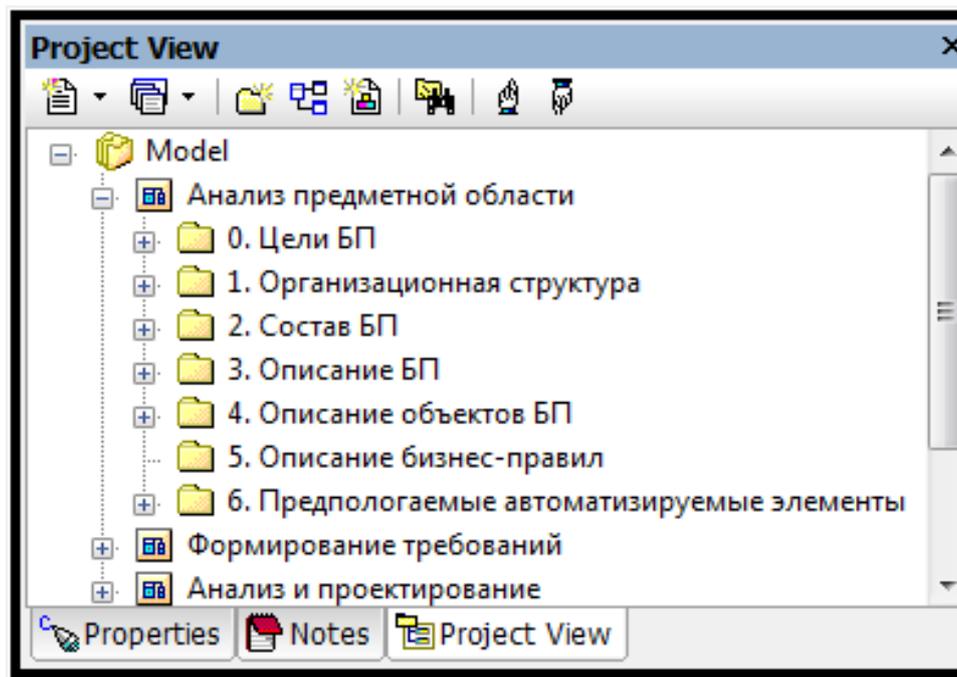


Рис. 2. Состав моделей при анализе предметной области

Модель «Цели бизнес-процессов» должна отображаться с помощью диаграммы классов (class diagram). Цели представляются в виде элементов «класс» со стереотипом «цель (target)».

Организационная структура может быть представлена в виде настраиваемой пользователем диаграммы (рис. 3) (custom diagram). Подразделения отображаются в виде пакетов. Работники подразделения (их должности) отображаются в виде элементов «class» со стереотипом «работник (internal worker)».

Состав бизнес-процессов можно отобразить в виде диаграммы бизнес-вариантов использования «business use case diagram» (рис. 4).

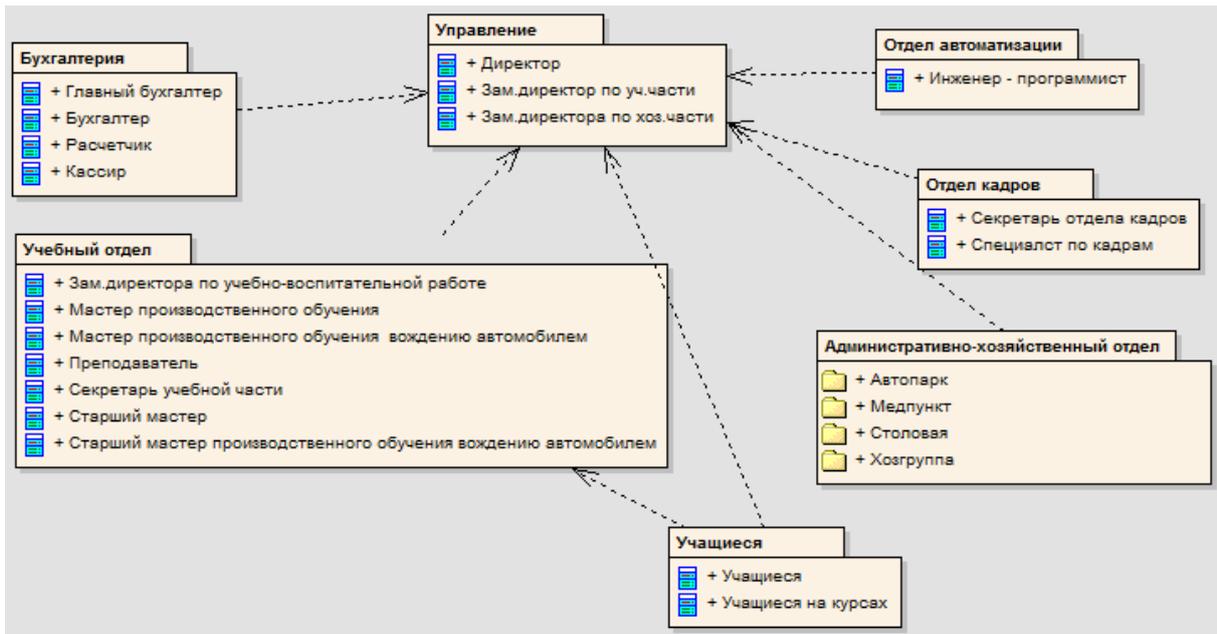


Рис. 3. Модель организационной структуры предприятия

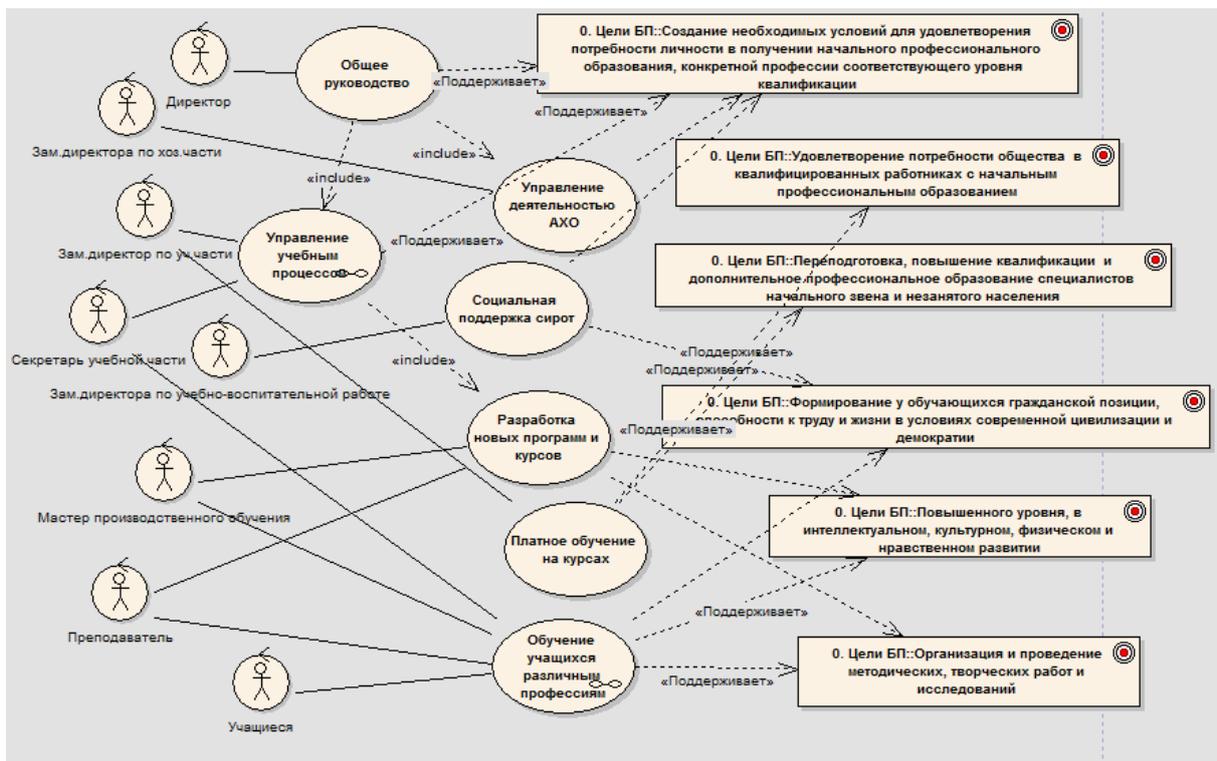


Рис. 4. Состав бизнес-процессов предприятия

Бизнес-процессы отображаются в виде элементов «use case», внешние сущности, инициирующие БП, в виде элементов «actor».

Содержание бизнес-процессов отображается в виде многоуровневых диаграмм деятельности (рис. 5). Верхний уровень диаграмм отображает макрошаги БП. Диаграммы нижнего уровня подробно описывают БП. На них указываются потоки объектов, то есть объекты, являющиеся входными и выходными для каждой деятельности. На диаграммах в отдельных дорожках необходимо отобразить исполнителей деятельностей (из структуры предприятия).

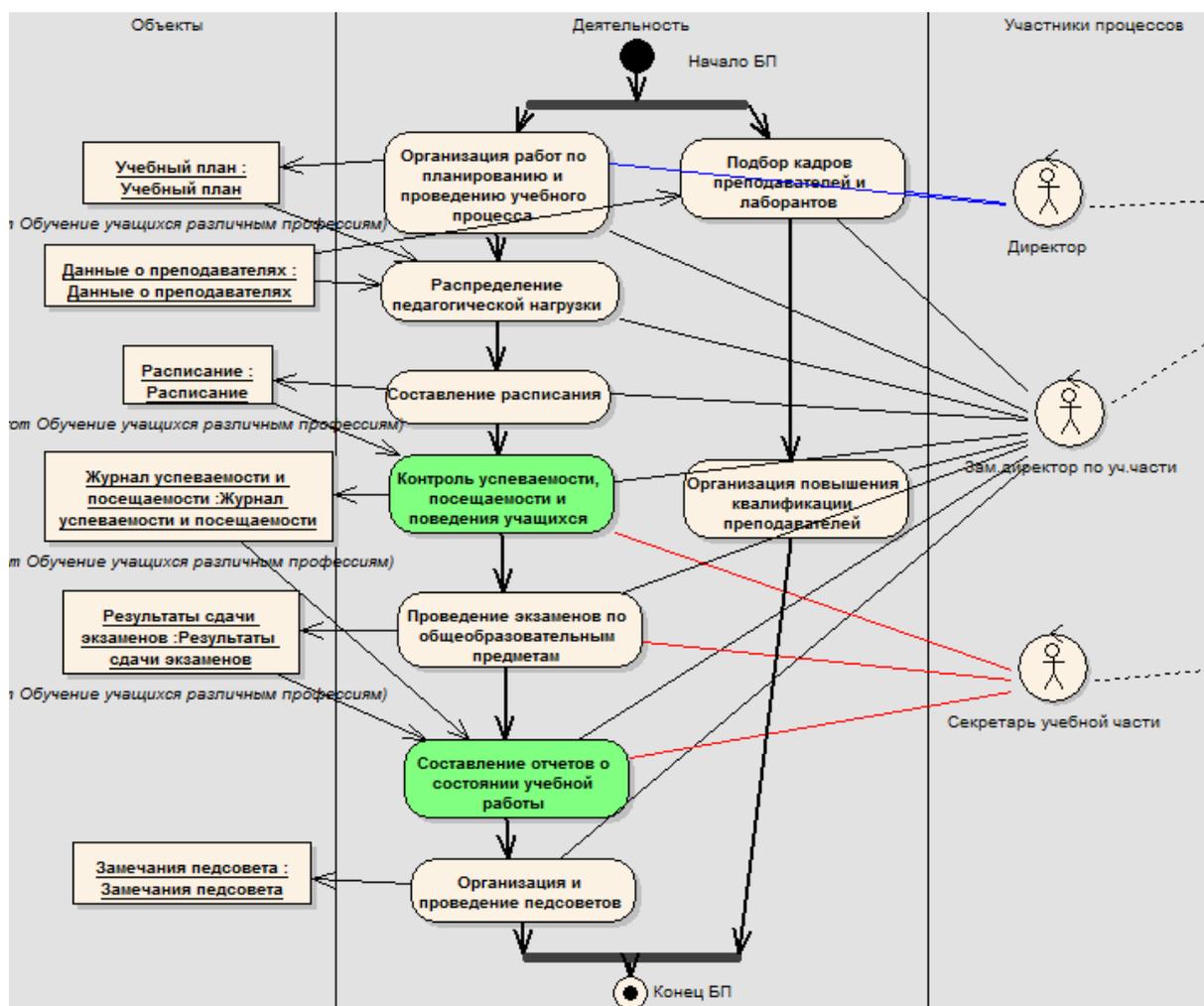


Рис. 5. Диаграмма содержания бизнес-процесса

Определение классов бизнес-объектов (КБО). КБО описывают материальные и информационные объекты реального мира, связанные с бизнес-процессами. Они представляют собой классы с атрибутами, операциями, обязанностями в соответствии с логикой работы предметной области.

Работу по выявлению КБО можно проводить параллельно с разработкой других моделей предметной области, особенно с разработкой диаграмм, отображающих содержание БП. Необходимо проанализировать объекты, участвующие в сценариях выполнения бизнес-процессов, выявить, какие из этих объектов можно отнести к одному классу, и для данной группы объектов ввести общий группирующий класс бизнес-объекта в виде класса разрабатываемой модели. Для КБО должны быть назначены атрибуты, операции, соответствующие их обязанностям в рассматриваемых сценариях БП.

После формирования КБО для каждого экземпляра объекта, участвующего в сценарии БП в виде потока объектов, должен быть назначен КБО в качестве классификатора, то есть объект-экземпляр сценария должен быть отнесен к КБО.

Состав КБО, их связи должны быть отображены на диаграмме классов (class diagram).

Автоматизируемые бизнес-решения (АБР). Работа по анализу условий заказчика завершается выявлением элементов бизнес-процессов, требующих автоматизации (автоматизируемые бизнес-решения). Автоматизируемые бизнес-решения соответствуют элементам БП, занимающим узловое положение в бизнес-процессах и автоматизация которых приведет к максимальному выигрышу. Такие элементы могут повторяться в нескольких БП. Автоматизируемые бизнес-решения включаются в модель в виде элемента UML «Требование (Requirement)» и отображаются на диаграмме пакетов.

Результатом процесса «анализ условий предприятия» является систематизированная информация, включающая:

- *название организации, ее статус, форма юридического лица (текстовое описание);*
- *ее масштаб (годовой оборот, доход, количество сотрудников и т.д.) (текстовое описание);*
- *общие принципы функционирования организации (текстовое описание);*
- *вид и профиль деятельности (текстовое описание);*
- *цель функционирования (диаграмма «Цели бизнес-процессов» с текстовым пояснением, раскрывающим цели);*

- *описание структуры организации, связанной с выполнением поставленной задачи (диаграмма «организационная структура предприятия»), с пояснением, раскрывающим назначение подразделений и функций сотрудников;*
- *описание состава бизнес-процессов предприятия, представленное в виде диаграмм «бизнес-вариантов использования» (см. рис. 4) с их кратким описанием. В описание должна включаться характеристика бизнес-актеров, использующих БП;*
- *описание содержания бизнес-процессов. Представляется в виде диаграмм деятельности. Для каждой диаграммы деятельности разрабатывается описание, включающее описание действий, выполняемых в отдельных деятельности (activity), объектов, используемых в деятельности, сотрудников, выполняющих деятельности;*
- *описание классов-объектов предметной области, выявленных для объектов сценариев бизнес-процессов. Классы объектов предметной области должны быть представлены в виде диаграммы классов с пояснением выявленных классов (название, роль, которую играет класс, атрибуты класса);*
- *описание автоматизируемых бизнес-решений. Для каждого автоматизируемого бизнес-решения необходимо сформулировать, какому действию, функции бизнес-процесса оно соответствует, его роль в бизнес-процессе и в достижении цели организации, какую выгоду принесет автоматизация данного типового решения.*

4.1.3. Формирование требований к системе

Для выполнения этой работы необходимо:

- **определить состав функциональных требований;**
- **определить состав сценариев, реализующих функциональные требования и их варианты использования, построить диаграмму вариантов использования;**
- **разработать содержание сценариев, при этом предварительно выявить объекты, участвующие в сценариях;**
- **определить требования к пользовательскому интерфейсу;**
- **определить требования к хранилищу данных.**

Решение данных задач необходимо отобразить в виде соответствующих моделей: «состав требований», «диаграммы вариантов использования», «сценарии выполнения требований», «требования к пользовательскому интерфейсу». Диаграммы, отображающие данные модели, необходимо разместить в соответствующие пакеты области просмотра «формирование требований» (рис. 6).

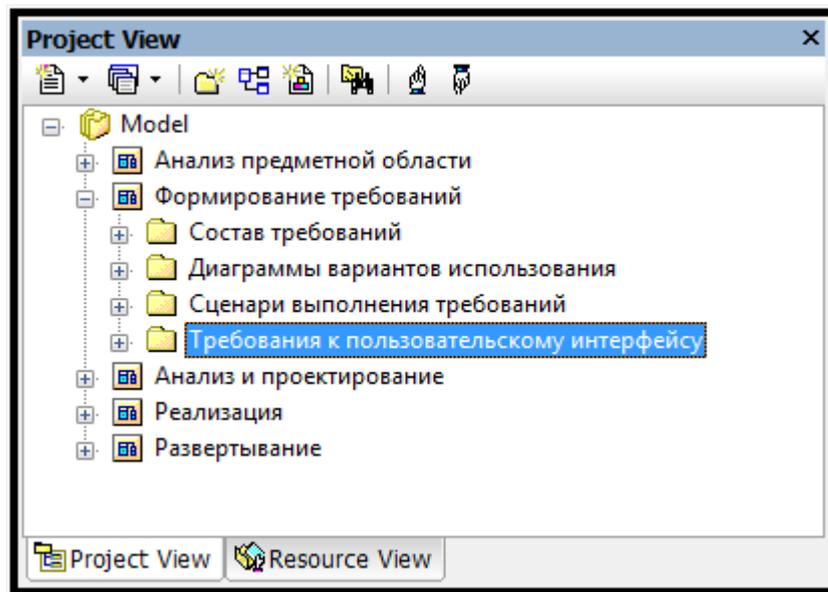


Рис. 6. Содержание моделей, разрабатываемых в процессе формирования требований

Данный этап проектирования необходимо оформить в виде группы пакетов, помещенных в отдельную область видимости «формирование требований».

Определение состава функциональных требований. Требования соответствуют выделенным в п. 4.1.2 автоматизируемым бизнес-решениям. Соответствие функциональных требований автоматизируемым бизнес-решениям можно отобразить в виде матрицы трассировки (рис. 7) и в виде диаграммы, на которой связь АБР и функциональных требований можно показать в виде отношения зависимости со стереотипом «трассировка (trace)».

№	Бизнес-процесс	Требование
1.	Зачисление студента	Зачисление студента
2.	Перевод студента	Перевод студента
3.	Отчисление студента	Отчисление студента
4.	Восстановление студента	Восстановление студента
5.	Выход из академического отпуска	Восстановление студента
6.	Подготовка отчетов	Подготовка отчетов
7.	Проведение сессии	Проведение сессии
8.	Ведение справочников	Ведение справочников
9.	Администрирование системы	Администрирование системы

Рис. 7. Матрица трассировки типовых решений БП в требования

В том случае, если требований много и они соответствуют выполнению различных функций, требования необходимо разделить на подсистемы (бизнес-подсистемы). На данном этапе подсистемы выделяются по функциональному признаку, то есть в одну подсистему объединяются однотипные функциональные требования, реализующие однотипные БП или связанные группы БП. Для отображения выявленных подсистем необходимо построить матрицу трассировки «подсистема – функциональные требования» (рис. 8) аналогично матрице, трассирующей АБР в требования.

Подсистема	Требование
Подсистема ведения личных дел	Зачисление студента
	Перевод студента
	Отчисление студента
	Восстановление студента
Подсистема ведения отчетов	Подготовка отчетов
	Проведение сессии
	Ведение справочников
Администрирование системы	Администрирование системы

Рис. 8. Матрица трассировки «подсистемы – функциональные требования»

Определение состава сценариев, реализующих требования. Для более точного определения функциональности разрабатываемого приложения необходимо сформировать состав ее сценариев, реализующих выявленные требования. В данном случае под сценарием подразумевается описание действий, заложенных в систему. Для реализации одного требования может быть использовано несколько сценариев системы. Состав разработанных сценариев, их связь с требованиями и с внешними сущностями отображается в виде *диаграммы вариантов использования* (рис. 9).

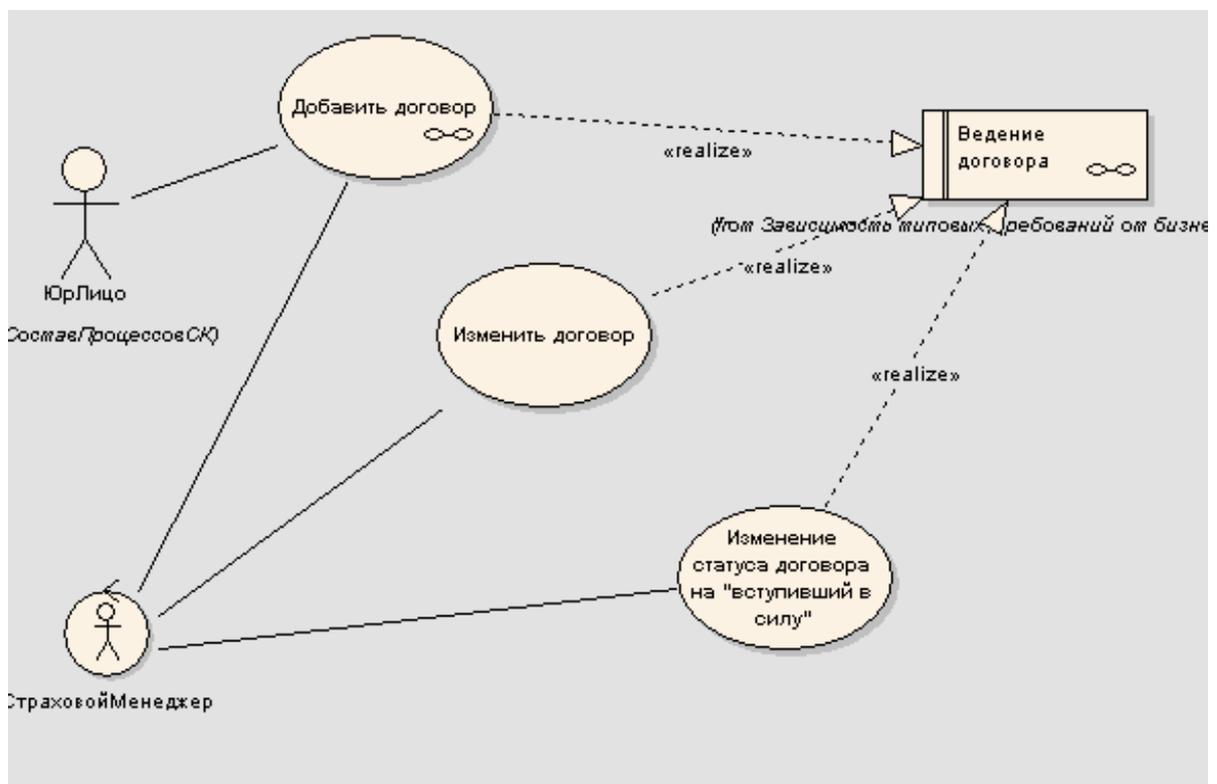


Рис. 9. Диаграмма вариантов использования, отображающая состав сценариев, реализующих функциональные требования

Состав сценариев отображается в виде элементов UML «use case», или «варианта использования». Каждый вариант использования подразумевает какую-то деятельность системы, вследствие которой «актант» получает ценный для него результат. Под актантом подразумевается некоторая внешняя сущность (реальный пользователь или внешняя система), инициирующая вариант использования или просто использующая его результаты. Вариант

использования связывается отношением зависимости со стереотипом «реализация (realize)» с соответствующим требованием.

Для каждого сценария необходимо сформировать его текстовое описание в виде возможных потоков событий, то есть действий, включаемых в него, обычно выделяется основной поток событий, альтернативный (используемый в штатном режиме, но при определенных условиях), и поток событий исключительный, обычно используемый при обработке ошибочной ситуации.

Сумма всех вариантов использования со связанными с ними актантами и поддерживаемыми функциональными требованиями составляет *модель вариантов использования*, которая описывает полную функциональность системы.

Разработка содержания сценариев. Определение содержания сценариев производится в виде декомпозиции соответствующих им «вариантов использования». Для каждого «варианта использования» формируется декомпозирующая диаграмма деятельности (рис. 10). Отличие ее от диаграммы, декомпозирующей БП, состоит в том, что на ней отображаются деятельности проектируемой системы, а не содержание бизнес-процесса. Отдельная деятельность (activity) может включать:

- ввод определенных данных в интерфейсные формы системы;
- сохранение данных;
- выполнение преобразования данных;
- другие действия, которые должна выполнять система, для выполнения функционального требования, которое реализует сценарий.

На диаграмме деятельности необходимо отобразить для каждого элемента-деятельности (activity) потоки входных и выходных объектов (object flow). Необходимо определить:

- какие данные вводятся в деятельность (activity) – отображается как объекты-сущности (стереотип объекта entity);
- с помощью какого объекта (формы) вводятся/выводятся данные – отображается как объекты-сущности (стереотип объекта boundary).

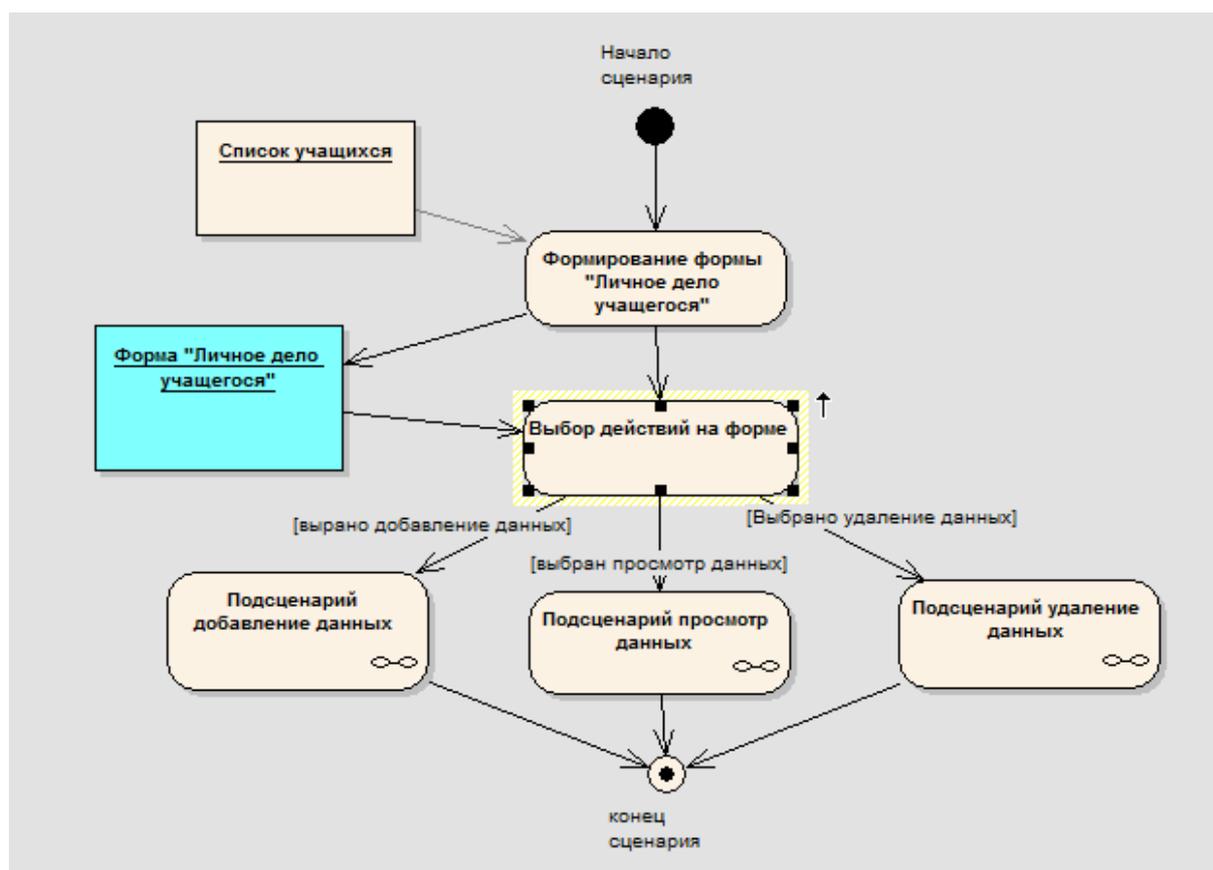


Рис. 10. Пример сценария выполнения требования

В некоторых случаях требуется введение специального контрольного объекта, выполняющего вспомогательные контрольные функции. Контрольный объект вводится, если рассматриваемая деятельность не связана непосредственно с выполнением бизнес-функции, а соответствует некоторым вспомогательным действиям, например, по контролю доступа, проверке прав, проверке правильности заполнения и др. Контрольный объект отображается с соответствующим стереотипом – *control*.

Определение требований к пользовательскому интерфейсу.

Требования к пользовательскому интерфейсу включают:

- его стиль;
- предпочтительную цветовую гамму;
- тип используемых управляющих элементов.

Более того, требования к интерфейсу должны включать предварительный состав экранных форм и их управляющих элементов.

Состав экранных форм определяется в процессе разработки сценариев, реализующих функциональные требования.

В данном пункте необходимо разработать предварительный эскиз форм, то есть определить их вид, состав управляющих элементов, их расположение (рис. 11). Для разработки эскиза форм можно воспользоваться средствами среды моделирования.

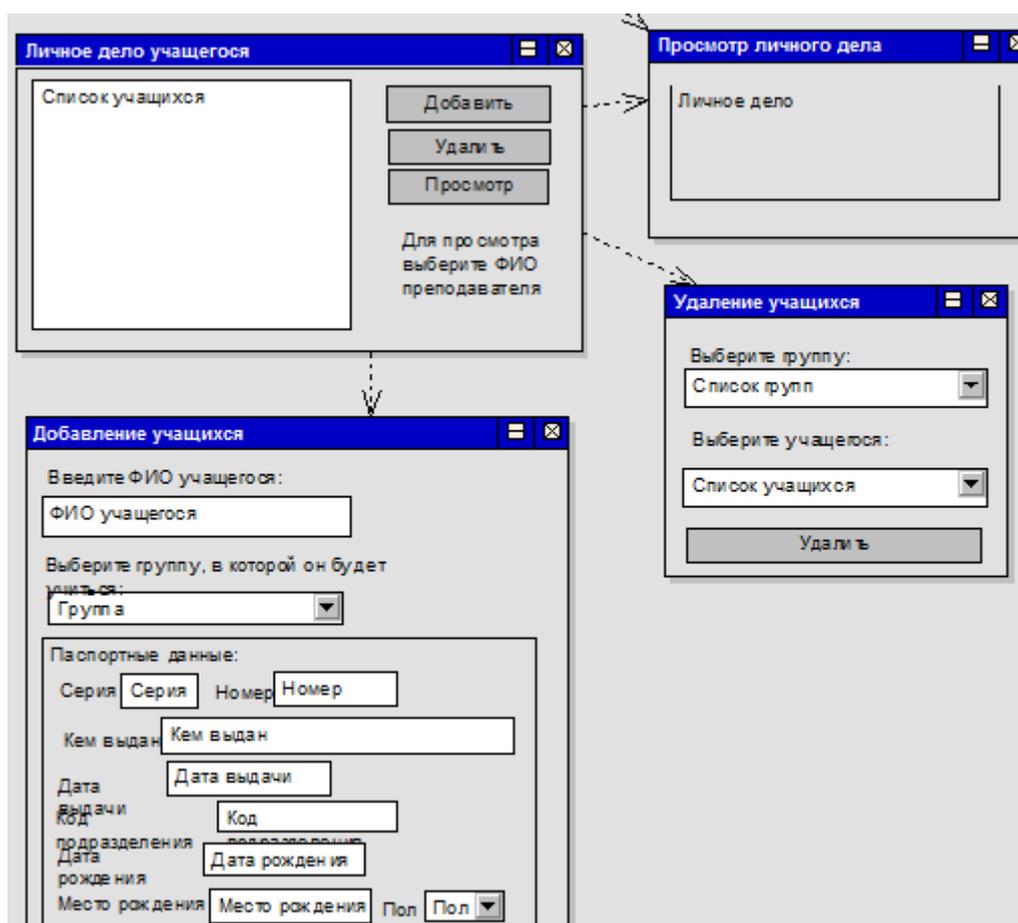


Рис. 11. Пример проекта пользовательского интерфейса

При выполнении данной работы необходимо стремиться, чтобы пользовательский интерфейс реализовывал выявленные бизнес-требования, то есть необходимо каждому бизнес-требованию сопоставить экранную форму.

Определение требований к хранилищу данных. Решение данной задачи выделяется в отдельный пункт, в связи ее спецификой.

Для проектирования базы данных необходимо, чтобы на уровне требований были определены особенности процесса хранения.

В качестве особенностей процесса хранения может быть задан тип используемой СУБД, ее архитектура, возможно предварительное определение состава информационных объектов БД. Вопросы, связанные с проектированием хранилища данных, можно рассматривать в соответствии с [3].

Результатом процесса формирования требований является систематизированная информация:

- *описание состава функциональных требований;*
- *диаграмма вариантов использования, отображающая функциональные требования к системе, и сценарии, их реализующие;*
- *диаграммы деятельности, декомпозирующие сценарии, реализующие функциональные требования. На данных диаграммах должны быть представлены потоки входных, выходных объектов-сущностей, граничные объекты. Диаграммы должны содержать описание, поясняющее выявленный состав объектов, и деятельности, выполняемые в сценарии;*
- *проект пользовательского интерфейса в виде диаграммы экранных форм;*
- *требования к хранилищу данных, описанные в виде отдельного пункта;*
- *документ, содержащий результирующее описание требований к системе (см. приложение).*

4.2. Разработка рабочего проекта

4.2.1. Общая характеристика выполняемых работ

На данном этапе разработки проводятся работы по выявлению объектов программной системы и их методов. Конкретное описание этих работ приведено в последующих пунктах.

4.2.2. Определение классов анализа

Классы анализа – это:

- граничные классы – классы объектов, реализующих формы;
- классы объектов сущностей – классы объектов, реализующих объекты-сущности;
- управляющие объекты – классы объектов, реализующих контрольные объекты.

Классы анализа системы определяются на основе объектов, выявленных при разработке сценариев, реализующих требования. То есть для объектов сценариев определяются классы. Кроме того, при формировании классов системы необходимо ориентироваться на классы объектов предметной области, выявленные в п. 4.1.2.

Результатом данной работы является систематизированная информация, включающая описание состава классов, их атрибутов, с указанием объектов, относящихся к данному классу.

4.2.3. Определение методов объектов

Методы объектов определяются на основе построения диаграмм последовательностей (рис. 12) для сценариев требований.

При построении диаграмм последовательностей каждое сообщение, направленное на объект, необходимо рассмотреть как метод данного объекта и включить как операцию, реализующую данный метод в соответствующий класс.

Результатом данной работы является систематизированная информация, включающая:

- *диаграммы последовательностей с описанием выявленных методов классов. Для каждого метода необходимо указать особенности его реализации;*
- *итоговая диаграмма классов, включая отношения между выявленными классами.*

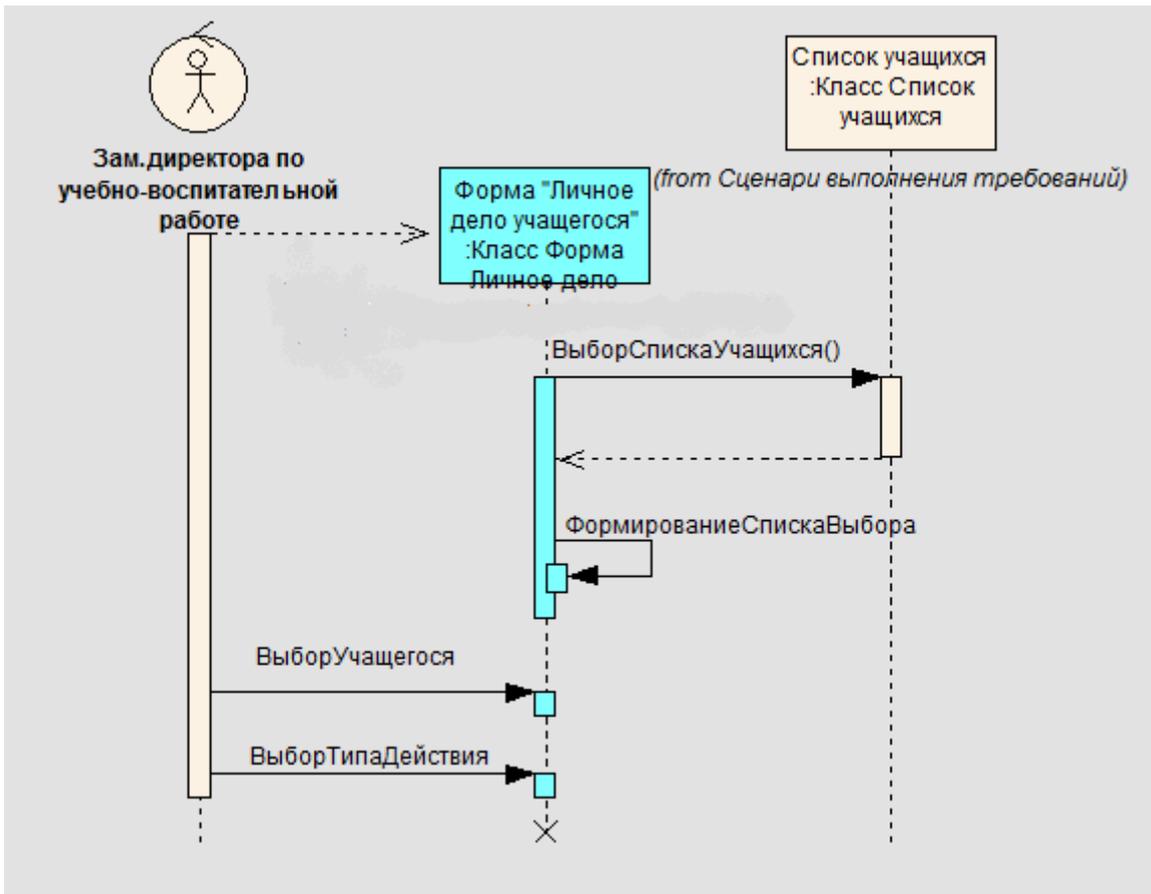


Рис. 12. Диаграмма последовательностей для сценария «Выбор учащегося и типа действия»

4.2.4. Выбор технологий реализации

К технологическим вопросам относятся:

- языки программирования;
- методы доступа к данным;
- стандарты взаимодействия компонентов;
- системные сервисы;
- операционные системы (в рамках которых возможна реализация требований, предъявляемых к ИС);
- параметры среды развёртывания.

При выборе технологий проводится повторный анализ требований к системе. Выделяются те требования, связанные с технологиями реализации. Эти требования могут быть связаны:

- с принципами взаимодействия пользователя с системой;
- с требуемым принципом доступа к данным;

- с принципом обмена информацией;
- с физическими ограничениями инфраструктуры;
- с физическими требованиями к решению.

Рассматриваются риски, связанные с реализацией данных требований в рамках существующей инфраструктуры.

При выборе операционной системы необходимо указать, в рамках какой операционной системы возможна реализация требований. Выбор операционной системы должен обеспечивать предполагаемое развитие системы, ее расширение и развертывание.

При выборе типа взаимодействия пользователя с системой, технологии пользовательского интерфейса определяется тип интерфейса. Обычно выбор состоит из двух альтернатив:

- Web-интерфейс;
- интерфейс в стиле операционной системы или родной интерфейс.

Если интерфейс в стиле операционной системы обеспечивает большие возможности для отображения пользовательских функций, лучшие характеристики по оперативности обмена данными с другими компонентами, то Web-интерфейс дает лучшие возможности по обновляемости пользовательской компоненты, обеспечивает удаленное взаимодействие компонент.

Выбор технологии взаимодействия пользовательских компонент с данными определяется выбранной операционной системой, методами и средствами хранения данных доступностью и квалификацией пользователей. В среде Windows основной технологией доступа к данным является технология ADO.NET, реализованная через соответствующие компоненты сред проектирования Visual Studio и Delphi Studio, однако возможно применение и других технологий при соответствующем обосновании.

Выбор языка программирования и среды разработки должен обосновываться наличием необходимой квалификации, опытом работы и доступностью лицензионных версий соответствующих продуктов.

При определении параметров среды развёртывания необходимо установить какого типа физические узлы будут использоваться для развёртывания приложения, характеристики этих узлов. Характеристики узлов должны согласовываться с требовани-

ями (ограничениями заказчика). Среда развёртывания должна быть представлена в виде начального варианта диаграммы развёртывания (рис. 13).

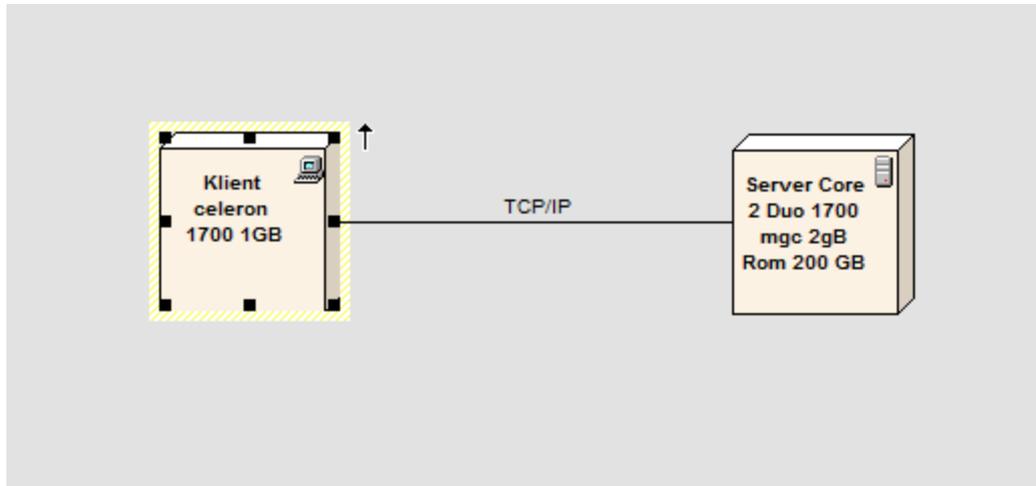


Рис. 13. Начальный вариант среды развёртывания.

4.2.5. Проектирование хранилища данных

В данном разделе происходит проектирование базы данных. Проектирование производится на основе требований, специально определенных для данного процесса. Кроме того, в состав отношений базы данных включаются объекты, для которых установлен атрибут «Persistent», то есть данный объект рассматривается как постоянный. В качестве постоянных классов включаются те, которые необходимо сохранять не только в процессе работы системы. Операции, связанные с обработкой данных объектов, можно оформлять как хранимые процедуры.

Результаты процесса «анализ и проектирование» должны быть отображены в виде моделей, помещенных в отдельные пакеты (рис. 14).

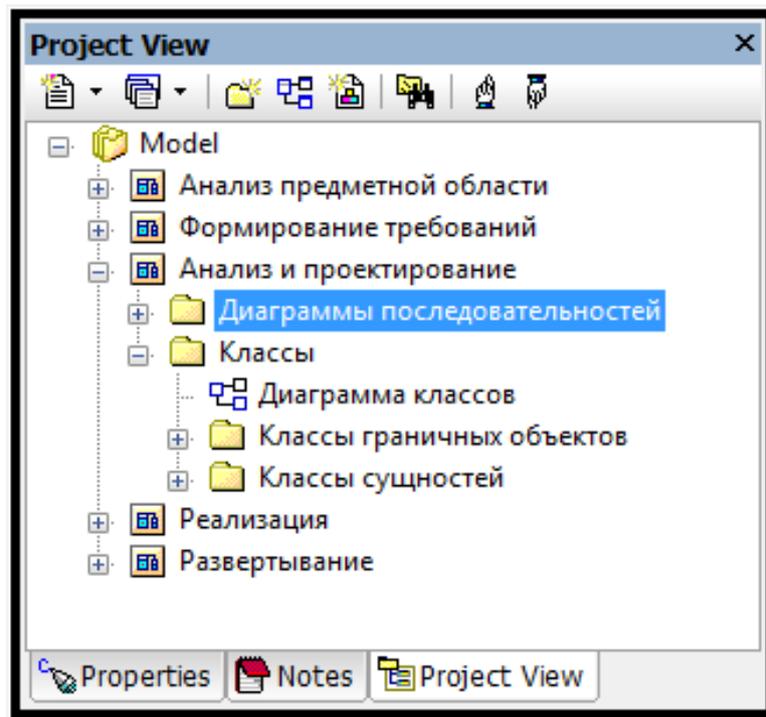


Рис. 14. Содержание моделей, разрабатываемых в процессе анализа и проектирования

Результатом данного этапа является систематизированная информация:

- *описание состава предполагаемых объектов базы данных, их атрибутов;*
- *диаграмма классов для классов объектов базы данных.*

4.3. Разработка сценариев тестирования

Для каждого сценария, реализующего требования, необходимо разработать сценарии тестирования. Сценарии тестирования выявляют наличие данного требования в системе.

4.4. Реализация проекта

Разработка программного кода может производиться автоматическим генерированием в среде проектирования с последующей его доработкой. Однако в любом случае требуется детальное описание полученных результатов процесса разработки и самого процесса.

Результат разработки программного кода должен быть представлен как модель реализации (рис.15).

Описание процесса реализации должно включать описание того каким образом были реализованы классы, выявленные на этапе проектирования.

Например:

Класс FORM1 – реализован в виде компонента Form1.cs и связанным с ним автоматически генерируемым файлом Form1.Desiner.cs .

Классы – сущности реализованы в виде соответствующих отношений базы данных.

Модель реализации включает объединенные в подсистемы компоненты, реализующие классы, выявленные в процессе проектирования.

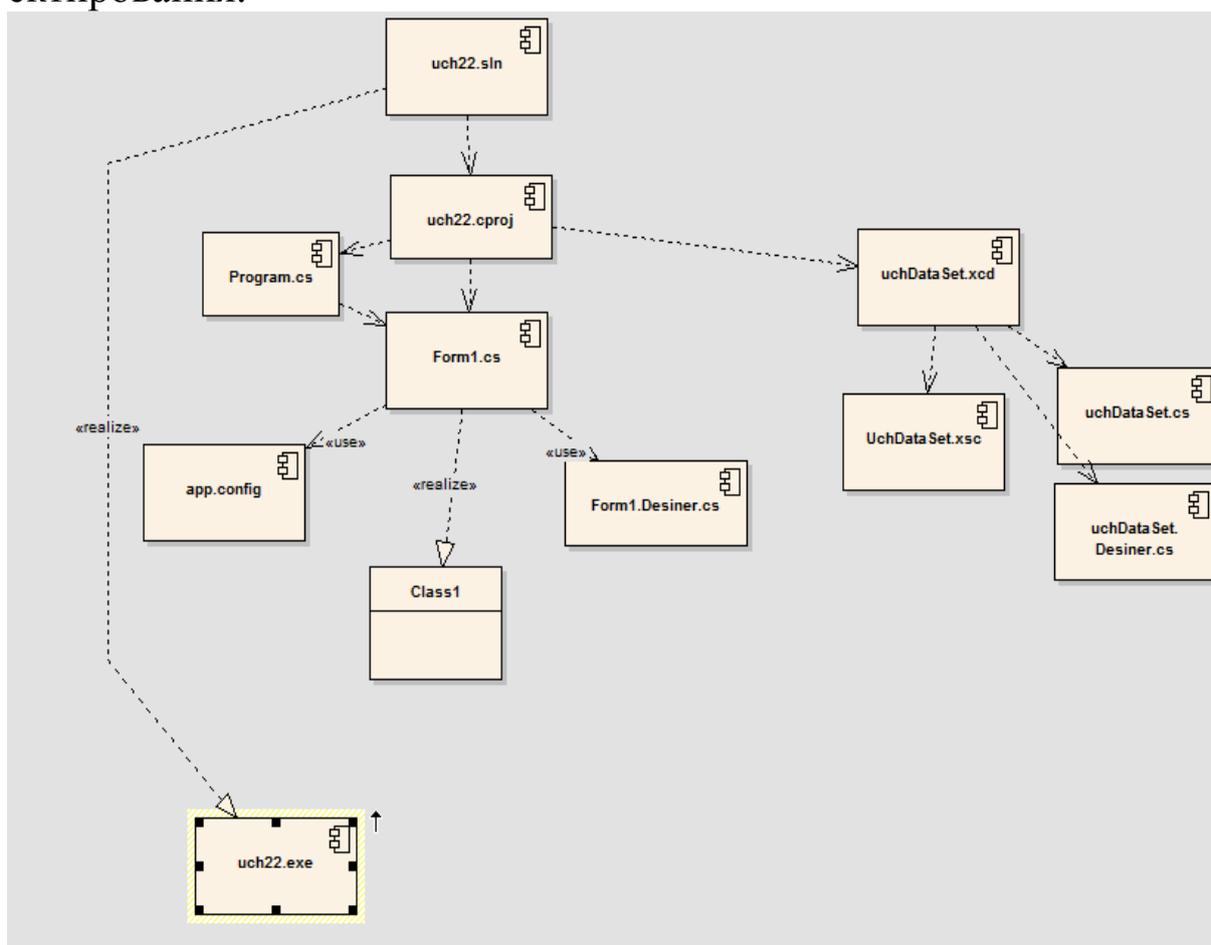


Рис. 15 Модель реализации

Для компонентов реализации должно быть сделано краткое описание. Описываться должны как исходные компоненты, на основе которых создаётся компонента, устанавливаемая как готовый продукт, в процессе развёртывания, так и сами результирующие компоненты развёртывания.

Например:

Uch22.sln – файл решения, содержит ссылку на файл проекта, определяет исходный объект для создаваемого управляемого модуля

Uch22.proj – содержит ссылки на компоненты, включаемые в проект и ссылки на используемые компоненты.

.....

Form1.cs – файл, содержащий описание класса Form1 (

Uch22.exe – файл полученный в результате компиляции исходных компонент.

Для каждой компоненты необходимо привести, каким образом она отображается на рабочем экране пользователя (скриншот формы в рабочем состоянии)

4.5. Развёртывание ИС

Описание процесса развёртывания (рис. 16) должно включать описание того какие компоненты, каким образом необходимо установить на рабочие узлы для работы созданной информационной системы. Иллюстрируется схемой развёртывания с указанием компонент, устанавливаемых на узлах. С соответствующим описанием

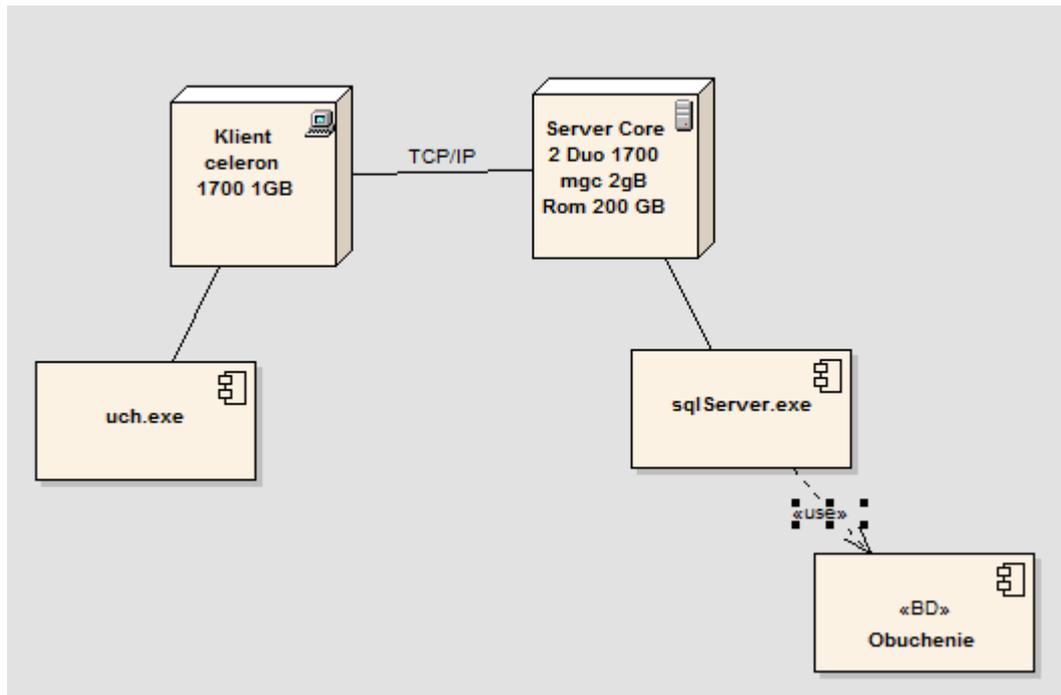


Рис. 16. Модель развёртывания.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Ларман, Крэг. Применение UML 2.0 и шаблонов проектирования. Практическое руководство / пер. с англ. – 3-е изд. – М. : ООО "ИД Вильяме", 2013. – 736 с. : ил. – Парал. тит. англ.
2. Принципы проектирования и разработки программного обеспечения : учебный курс MCSD / пер. с англ. – 2-е изд., испр. – М. : ИТД «Русская Редакция», 2002. – 736 с.
3. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч и др.; пер. с англ. – 3-е изд. – М.: ООО «ИД Вильямс», 2010. – 720 с.

Спецификация программных требований

1. Введение

Введение к документу «Спецификация программных требований» (SRS) должно дать его полный обзор. В него необходимо включить назначение документа, область применения, определения, акронимы, сокращения, ссылки и обзор рассматриваемой спецификации.

Компоновку спецификации SRS можно выполнять различными способами. Эти вопросы вместе с другими возможностями организации спецификации SRS более подробно раскрываются в документе IEEE830-1998.

1.1. Назначение

Цель создания данного документа.

«Спецификация программных требований» должна полностью описывать внешнее поведение определяемого приложения или подсистемы, нефункциональные требования, проектные ограничения и другие факторы, необходимые для полного и всестороннего описания требований к программному обеспечению.

Например: Данный документ предназначен для описания требований к проектируемой программной системе «Система автоматического анкетирования».

1.2. Область применения

Краткое описание программы, функции или группы подсистем, к которым относится данная спецификация. Указываются модели сценариев использования, с которыми она связана, и все прочие сведения, имеющие к ней отношение.

Например: Данная программная система будет применена в рамках системы менеджмента качества для проведения маркетинговых исследований среди потребителей (например, абитуриентов, студентов, выпускников) и сотрудников организации (например, образовательного учреждения) с целью определения степени удовлетворенности заинтересованных сторон.

1.3. Определения, акронимы и сокращения

В этом разделе даются определения всех терминов, акронимов и сокращений, которые необходимы для правильной интерпретации документа «Спецификация программных требований». Данная информация может быть представлена в виде ссылки на словарь проекта.

1.4. Ссылки

В этом разделе дается полный список всех документов, на которые ссылается «Спецификация программных требований». Каждая ссылка должна содержать заголовок документа, номер отчета (при наличии), дату и название организации, опубликовавшей документ. Следует указать также, из каких источников могут быть получены эти сведения. Эта информация может быть представлена в виде ссылки на приложение или на другой документ.

1.5. Обзор

Дается описание остальной информации из «Спецификации программных требований» и структуры этого документа.

2. Общее описание

Здесь следует описать общие факторы, влияющие на продукт и требования к нему. Конкретные требования подробно определяются в разд. 3, а в этом разделе необходимо дать лишь обоснование, которое поможет их понять. В частности, рассматриваются следующие вопросы:

- перспектива продукта;*
- функции продукта;*
- пользовательские характеристики;*
- ограничения;*
- предположения и зависимости;*
- подмножества требований.*

3. Конкретные требования

Этот раздел спецификации SRS должен содержать все программные требования. Уровень детализации должен быть достаточным для проектирования системы и ее тестирования на соответствие этим требованиям. В случае применения моделирования сценариев использования данные требования фиксируются в этих сценариях и в соответствующем документе дополнительных технических требований. В противном случае

набросок дополнительных технических требований можно вставить непосредственно в данный раздел, как показано ниже.

3.1. Функциональность

В этом разделе описываются функциональные требования к системе, выражаемые естественным языком. Для многих приложений эти сведения составляют пакет спецификаций требований к системе (SRS), поэтому следует хорошо продумать структуру этого раздела. Как правило, этот раздел организуется по реализуемым функциям, однако можно применить и альтернативные методы, например, организацию по пользователям или подсистемам. Функциональные требования могут включать в себя списки реализуемых функций, возможностей и средств обеспечения безопасности данных.

Если для выяснения функциональных возможностей используются средства разработки приложений, например, инструменты управления требованиями или моделирования, то данный раздел должен содержать ссылки на доступные источники этих данных с указанием местонахождения и названия используемого инструмента.

Например:

3.1.1. Система должна обеспечивать проведение анкетирования различных групп заинтересованных сторон на основе заполнения электронных опросных форм, где представлены группы вопросов с возможными вариантами ответов, открытые вопросы (т.е. вопросы без вариантов ответов) и вопросы с возможностью добавления собственных ответов, ранее не зарегистрированных в системе.

3.1.2. Анкетируемым присваивается внутренний идентификационный код, кроме того, анкетируемый должен ввести информацию, характеризующую его социальный статус (пол, специальность, факультет и т.д.). Данная информация запоминается.

3.1.3. Система должна обеспечивать следующие административные функции.

3.1.3.1. Автоматическое формирование опросных форм на основе представленных в анкетных листах вопросах с учетом пожеланий заказчика о представлении вопросов и ответов на электронных формах.

3.1.3.2. Возможность блокирования последующих вопросов в зависимости от ответа на предыдущие вопросы.

3.1.3.3. Корректировка состава вопросов и возможных для них ответов.

3.1.3.4. Возможность анализа вариантов ответов, внесенных пользователями, с последующим их отсеком или включением в список возможных вопросов.

3.1.3.5. Должна производить анализ ответов.

3.1.3.5.1. Распределение по частотам различных ответов на определенные вопросы.

3.1.3.5.2. Сравнение результатов опросов для различных типов пользователей.

3.2. Практичность

В этот раздел следует включить все требования, имеющие отношение к практичности. Например:

- *укажите необходимое время обучения обычных и квалифицированных пользователей, которое позволит им достичь высокой производительности при выполнении определенных операций;*

- *укажите измеряемые показатели времени выполнения для характерных задач или обоснуйте требования к практичности новой системы на примере других систем, хорошо знакомых пользователям;*

- *укажите требования общих стандартов практичности, например, стандартов общего пользовательского доступа (CUA) корпорации IBM или стандартов графического пользовательского интерфейса корпорации Microsoft.*

3.2.1. Первое требование к практичности

Описание требования.

3.3. Надежность

Здесь должны быть описаны требования к надежности системы. Предполагается следующий вид раздела:

- *доступность – указывается процент времени, в течение которого система будет доступна (xx.xx%), предполагаемые часы использования, доступ для технического обслуживания, режим работы при пониженной производительности и т.д.;*

- *среднее время безотказной работы (MTBF) – обычно указывается в часах, но может быть выражено в днях, месяцах или годах;*
- *среднее время устранения неисправностей (MTTR) – указывается допустимое время простоя системы после сбоя;*
- *точность – приводятся показатели четкости (разрешения) и точности (по какому-либо известному стандарту), которые требуются на выходе системы;*
- *максимально допустимое количество ошибок – обычно выражается в количестве ошибок на тысячу строк кода (KLOC) либо на количество реализуемых функций;*
- *уровень ошибок – определение категорий незначительных, существенных и грубых ошибок. Требования должны определять, что подразумевается под грубой ошибкой (например, полная потеря данных или полная неспособность использовать какие-либо функциональные возможности системы).*

3.3.1. Первое требование к надежности

Описание требования.

3.4. Производительность

В этом разделе должны быть кратко изложены основные характеристики производительности системы. Укажите характерные показатели для времени реакции. Там, где это возможно, сошлитесь на имена соответствующих сценариев использования:

- *время реакции для операции (среднее, максимальное);*
- *производительность (например, число транзакций в секунду);*
- *полная мощность (например, число заказчиков или операций, которые может обслуживать система);*
- *режимы снижения производительности (опишите приемлемые режимы работы, при которых производительность системы каким-либо образом снижается);*
- *использование ресурсов: память, дисковое пространство, связь и т.д.*

3.4.1. Первое требование к производительности

Описание требования.

3.5. Возможности поддержки

В этом разделе следует описать все требования, которые способны усилить возможности технической поддержки создаваемой системы, включая стандарты программирования, соглашения о присвоении имен, библиотеки классов, доступ и средства технического обслуживания.

3.5.1. Первое требование к возможностям поддержки

Описание требования.

3.6. Проектные ограничения

В этом разделе описываются все проектные ограничения, применяемые к создаваемой системе. Эти ограничения представляют собой проектные решения, которых необходимо твердо придерживаться. Примерами могут служить языки программирования, требования к программному процессу, использование определенных средств разработки, архитектурные и проектные ограничения, приобретаемые компоненты, библиотеки классов и т.д.

3.6.1. Первое проектное ограничение

Описание требования.

3.7. Требования к интерактивной пользовательской документации и справочной системе

Опишите существующие требования к интерактивной пользовательской документации, справочным системам, сведениям о продукте и т.д.

3.8. Приобретаемые компоненты

В этом разделе следует описать все приобретаемые компоненты системы, все применяемые лицензионные нормы или ограничения использования, а также все связанные с этими компонентами стандарты совместимости, взаимодействия или интерфейсов.

3.9. Интерфейсы

В этом разделе определяются интерфейсы, которые должны поддерживаться данным приложением. Описание должно содержать их соответствующие характерные особенности, протоколы, порты, логические адреса и другие сведения, позволяющие контролировать соответствие разрабатываемого программного обеспечения требованиям к его интерфейсам.

3.9.1. Пользовательские интерфейсы

Опишите пользовательские интерфейсы, реализуемые в данном программном обеспечении.

3.9.2. Аппаратные интерфейсы

В этом разделе следует определить все аппаратные интерфейсы, поддерживаемые данным программным обеспечением, включая их логическую структуру, физические адреса, ожидаемое поведение и т.д.

3.9.3. Программные интерфейсы

В этом разделе следует описать программные интерфейсы между данным приложением и другими компонентами программной системы. В число этих компонентов могут входить приобретаемые модули, повторно используемые компоненты других приложений или же те компоненты, которые разрабатываются для подсистем вне функциональных границ данной спецификации программных требований, но с которыми это приложение должно взаимодействовать.

3.9.4. Коммуникационные интерфейсы

Опишите все коммуникационные интерфейсы, связанные с другими системами или устройствами, такими как локальные сети, удаленные устройства и т.д.

3.10. Лицензионные требования

Опишите все обязательные лицензионные требования или другие ограничения использования, которые накладываются на программное обеспечение.

3.11. Предупреждения, касающиеся законодательства, авторских прав, и другие замечания

В этом разделе следует описать все необходимые юридические предупреждения, гарантии, авторские и патентные права, текстовые торговые марки, знаки или логотипы, которые имеют отношение к данному программному обеспечению.

3.12. Применяемые стандарты

В этом разделе должны быть приведены ссылки на все стандарты и их конкретные разделы, применяемые к описываемой системе. В их число могут входить, например, юридические стандарты, стандарты качества и стабильности, отраслевые стандарты практичности, взаимодействия, локализации, совместимости с операционными системами и т.д.

4. Сопровождающая информация

Сопровождающая информация упрощает использование Спецификации программных требований. В нее входят:

- *содержание;*
- *алфавитный указатель;*
- *приложения.*

*Сюда могут входить детализированные описания сценариев использования или прототипы пользовательских интерфейсов. В случае использования приложений в спецификации **SRS** необходимо четко установить, являются ли эти приложения частью требований.*

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	1
2. ТЕМА КУРСОВОГО ПРОЕКТА	2
3. ФОРМА ПРЕДСТАВЛЕНИЯ ПРОЕКТНЫХ РЕШЕНИЙ И ИСПОЛЬЗУЕМЫЕ ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА	2
4. СОДЕРЖАНИЕ КУРСОВОГО ПРОЕКТА	3
4.1. Разработка технического задания	3
4.1.1. Исходная постановка задачи проектирования	4
4.1.2. Анализ условий предприятия	4
4.1.3. Формирование требований к системе	9
4.2. Разработка рабочего проекта	16
4.2.1. Общая характеристика выполняемых работ	16
4.2.2. Определение классов анализа	17
4.2.3. Определение методов объектов	17
4.2.4. Выбор технологий реализации	18
4.2.5. Проектирование хранилища данных	20
4.3. Разработка сценариев тестирования	21
4.4. Реализация проекта	21
4.5. Развёртывание ИС	23
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ	24
ПРИЛОЖЕНИЕ	25
Спецификация программных требований	25
1. Введение	25
1.1. Назначение	25
<i>Например:</i> Данный документ предназначен для описания требований к проектируемой программной системе «Система автоматического анкетирования»	25
1.2. Область применения	25
1.3. Определения, акронимы и сокращения	26
1.4. Ссылки	26
1.5. Обзор	26
2. Общее описание	26
3. Конкретные требования	26
3.1. Функциональность	27
3.2. Практичность	28
3.2.1. <i>Первое требование к практичности</i>	28
3.3. Надежность	28
3.3.1. <i>Первое требование к надежности</i>	29
3.4. Производительность	29
3.4.1. <i>Первое требование к производительности</i>	29
3.5. Возможности поддержки	30
3.5.1. <i>Первое требование к возможностям поддержки</i>	30
3.6. Проектные ограничения	30
3.6.1. <i>Первое проектное ограничение</i>	30
3.7. Требования к интерактивной пользовательской документации и справочной системе	30
3.8. Приобретаемые компоненты	30
3.9. Интерфейсы	30
3.9.1. <i>Пользовательские интерфейсы</i>	31
3.9.2. <i>Аппаратные интерфейсы</i>	31
3.9.3. <i>Программные интерфейсы</i>	31
3.9.4. <i>Коммуникационные интерфейсы</i>	31
3.10. Лицензионные требования	31
3.11. Предупреждения, касающиеся законодательства, авторских прав, и другие замечания	31
3.12. Применяемые стандарты	31
4. Сопровождающая информация	32