

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет
им. Г. И. Носова»
Многопрофильный колледж

**Методические указания
по выполнению и защите
дипломного проекта
для обучающихся
специальности 09.02.07 Информационные системы и
программирование
квалификация: разработчик веб и мультимедийных приложений**

Методические указания разработаны на основе ФГОС СПО по специальности 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от 09.12.2016 г. № 1547, СМК-К-О-СМГТУ-2/2-6-26 Инструкция по оформлению курсового / дипломного проекта (работы) по образовательным программам среднего профессионального образования.

Разработчик:

преподаватель отделения №2 «Информационных технологий и транспорта»

Многопрофильного колледжа ФГБОУ ВО «МГТУ им. Г.И. Носова»

Ирина Геннадьевна Зорина

СОДЕРЖАНИЕ

1	Общие положения и организация выполнения дипломного проекта	4
2	Требования к структуре дипломного проекта	6
3	Особенности построения отдельных разделов дипломного проекта	8
	3.1.1 особенности построения теоретико-информационного анализа предметной области	8
	3.1.1.1 анализ предметной области	8
	3.1.1.2 анализ и оценка существующих программных продуктов	
	3.1.1.3 постановка задачи	9
	3.1.2 особенности построения части «проектирование программного продукта»	9
	3.1.2.1 моделирование программного продукта	10
	3.1.2.2 обоснование выбора средств разработки	15
	3.1.2.3 разработка базы данных	16
	3.1.2.4 логическая и физическая структура сайта	19
	3.1.2.5 проектирование меню и интерфейса программного продукта	20
	3.1.2.6 описание запросов при разработке программного продукта	
	3.1.2.7 проектирование отчетов	23
	3.1.2.8 администрирование программного продукта	24
	3.1.3 особенности построения части «описание работы программного продукта»	25
	3.1.3.1 руководство пользователя	25
	3.1.3.2 руководство программисту	26
	3.1.4 особенности построения раздела информационная безопасность	27
	3.1.4.1 виды возможных угроз программного продукта	27
	3.1.4.2 информационная безопасность программного продукта	27
	3.1.5 особенности построения части «управление проектом».	28
4	Порядок защиты дипломного проекта	45
	4.1 подготовка доклада для защиты	45
	4.2 подготовка презентации на защите	46

1 ОБЩИЕ ПОЛОЖЕНИЯ И ОРГАНИЗАЦИЯ ВЫПОЛНЕНИЯ ДИПЛОМНОГО ПРОЕКТА

Дипломный проект - итоговая аттестационная работа обучающегося, выполненная им на выпускном курсе, оформленная в письменном виде с соблюдением необходимых требований и представленная по окончании обучения к защите перед государственной экзаменационной комиссией, является обязательным аттестационным испытанием выпускника, завершающего обучение по специальности СПО 09.02.07 Информационные системы и программирование, и выполняется в виде дипломного проекта.

Дипломный проект – это выпускная работа обучающегося по программам технического профиля на соискание квалификации по специальности среднего профессионального образования. Представляет собой решение конкретной инженерной задачи по специальности. Оформляется в виде чертежей и пояснительной записки. К дипломному проекту могут прилагаться расчетно-графические материалы, программные продукты, рабочие макеты, материалы научных исследований и другие материалы, разработанные выпускником.

Дипломный проект является самостоятельной работой обучающегося, на основании которой Государственная экзаменационная комиссия (далее - ГЭК) решает вопрос о присвоении выпускнику квалификации – разработчик веб и мультимедийных приложений.

Защита дипломного проекта как форма государственной итоговой аттестации проводится с целью выявления соответствия уровня и качества подготовки выпускника требованиям Федерального государственного образовательного стандарта, установления уровня подготовки выпускника к выполнению профессиональных задач и соответствия его подготовки требованиям программы подготовки специалистов среднего звена.

Дипломный проект по специальности 09.02.07 Информационные системы и программирование направлен на а систематизацию и закрепление знаний выпускника по специальности, а также определение уровня готовности выпускника к самостоятельной профессиональной деятельности, развитие навыков ведения самостоятельной работы; овладение методиками научного исследования и экспериментирования; определение уровня готовности выпускника к самостоятельной профессиональной деятельности в условиях современного производства, прогресса науки и техники, углубленное изучение технологий разработки программных продуктов, программирования с использованием современных средств и овладение навыками самостоятельного решения инженерных задач

Дипломный проект предполагает самостоятельную подготовку (написание) выпускником проекта, демонстрирующего уровень знаний выпускника в рамках выбранной темы, а также сформированность его профессиональных умений и навыков.

Выполнение дипломного проекта состоит из нескольких этапов:

- выбор и закрепление объекта преддипломной практики;
- выбор и закрепление темы дипломного проекта;
- разработка и утверждение задания на дипломный проект;
- сбор материала для дипломного проекта на объекте преддипломной практики;
- написание и оформление пояснительной записки и презентации;
- предварительная защита дипломного проекта;
- защита дипломного проекта на заседании ГЭК.

Выполненный дипломный проект, подписанный выпускником и консультантами проходит процедуру нормоконтроля.

Нормоконтроль – процесс, осуществляющий выполнение норм, правил и требований, установленных в стандартах и другой нормативно-технической документации при разработке студентами дипломных проектов. Нормоконтроль дипломных проектов является завершающим этапом выполнения дипломного проекта.

При обнаружении ошибок, небрежного выполнения работы, отсутствии обязательных подписей, несоблюдении требований нормоконтролер возвращает выпускнику дипломный проект на исправление. Без подписи нормоконтролера дипломный проект к защите не допускаются.

2 ТРЕБОВАНИЯ К СТРУКТУРЕ ДИПЛОМНОГО ПРОЕКТА

2.1 В общем случае дипломный проект должен содержать:

- текстовый документ (пояснительную записку);
- графический материал.

2.2 Текстовый документ должен включать в указанной последовательности следующие элементы:

- титульный лист;
- задание;
- отзыв руководителя;
- содержание;
- введение;
- основная часть (разделы, подразделы, пункты в соответствии с утвержденным заданием);

- список использованных источников;
- приложения.

2.3 К графическому материалу следует относить:

- демонстрационные листы (плакаты);
- электронные презентации;
- чертежи и схемы.

2.4 Демонстрационные листы с графиками, фотографиями, схемами, чертежами представляются на листах формата А1. Объем графического материала определяется заданием и условиями защиты работы.

2.5 Работа, наряду с бумажным носителем, должна быть полностью представлена на электронных носителях

К графическому материалу следует относить электронные презентации.

Объем графического материала определяется заданием и условиями защиты работы (8-15 слайдов).

Работа, наряду с бумажным носителем, должна быть полностью представлена на электронных носителях.

Объем записки должен составлять 60-90 страниц печатного текста.

Структура дипломного проекта проектного характера

Содержанием дипломного проекта проектного характера является разработка продукта творческой деятельности. По структуре данный дипломный проект состоит из пояснительной записки, практической части и списка литературы.

В пояснительной записке дается теоретическое обоснование создаваемых продуктов творческой деятельности. Структуру и содержание пояснительной записки определяют в зависимости от

профиля специальности и темы дипломного проекта. Объем пояснительной записки должен составлять от 60 до 90 страниц печатного текста.

В практической части созданные продукты творческой деятельности представляется в виде серий наглядных пособий, компьютерных обучающих программ, в соответствии с видами профессиональной деятельности и темой дипломного проекта.

3 ОСОБЕННОСТИ ПОСТРОЕНИЯ ОТДЕЛЬНЫХ РАЗДЕЛОВ ДИПЛОМНОГО ПРОЕКТА

3.1 Основная часть.

3.1.1 Особенности построения теоретико-информационного анализа предметной области

Задачами теоретико-информационного анализа являются рассмотрение существующего состояния предметной области, характеристики объекта и системы управления и обоснование предложений по устранению выявленных недостатков, внедрению новых подходов, новых технологий защиты информации и т.д.

3.1.1.1 Анализ предметной области

В этом разделе следует описать деятельность предприятия/объекта, для которых разрабатывается проект. Перечислить функции, выполняемые на этом предприятии/объекте, круг решаемых задач, структуру предприятия, характеристику объекта. Предоставить описание деятельности структурных подразделений/объекта.

Информация собирается с учетом тех задач, которые поставлены в дипломном проекте.

3.1.1.2 Анализ и оценка существующих программных продуктов

Необходимо подробно и последовательно описать процесс решения задачи существующим в данный момент способом. Здесь описывается функциональная и информационная структура существующей системы, качественные и количественные характеристики, вскрывающие её компоненты и их взаимодействие. В описании показать последовательность процессов обработки данных, указать, какие входные и выходные документы задействованы. Обратит внимание на то, каким образом формируются выходные документы, какие расчеты, выборки, сортировки и другие операции над данными при этом производятся в существующей системе. При описании постараться разбить весь процесс на последовательность отдельных законченных этапов.

Внимательно изучив существующую систему, определить недостатки или неудобства, возникающие при решении задачи таким способом. Каждый недостаток описать конкретно для своей системы. Например, возможное возникновение вычислительных ошибок при расчетах определенных показателей, ненадежное хранение конкретных данных на бумажных носителях, большой документооборот при составлении сводного отчета или ведомости, длительный и не всегда успешный поиск нужной информации и т. д.

Описание существующих программных решений должно сопровождаться скринами интерфейса.

Перечислить предлагаемые мероприятия по усовершенствованию системы путем автоматизации, которые устранят перечисленные ранее

недостатки.

В технико-экономическом обосновании модернизации показать, какие этапы учёта для этого планируется автоматизировать. В каждом предлагаемом случае указать преимущества новой системы по сравнению с предыдущей.

Указать, с какой целью ведется разработка системы, условия ее использования на объекте автоматизации, ограничения, которые включены в систему по ряду причин: возможно, не рассмотрены какие-то определенные стороны учета или сужен круг рассматриваемых данных.

Указать требования к структуре и функционированию системы, к численности квалифицированных работников, к надежности и безопасности работы системы, к эргономике и эстетике, эксплуатации и техническому обслуживанию, к защите от несанкционированного доступа, требования по сохранности информации при авариях, к защите от внешней среды, к патентной чистоте проектных решений, требования по унификации и стандартизации.

Перечислить функции (функциональные подсистемы), которые будут реализованы в разрабатываемой системе. При определении набора функций системы можно воспользоваться одним из существующих принципов выделения функциональных подсистем: предметным, функциональным, смешанным (предметно-функциональным), проблемным. Функция - это законченный процесс обработки информации с определённым набором входных и выходных данных.

3.1.1.3 Постановка задачи

В разделе «Постановка задачи» указать цель и задачи разработки программного продукта, планируемый функционал, роли пользователей с доступными функциями.

3.1.2 Особенности построения части «Проектирование программного продукта»

Проектирование программного обеспечения – процесс создания проекта программного обеспечения.

Целью проектирования является определение внутренних свойств системы и детализации её внешних (видимых) свойств на основе выданных заказчиком требований к ПО (исходные условия задачи). Эти требования подвергаются анализу.

В зависимости от класса создаваемого ПО, процесс проектирования может обеспечиваться как «ручным» проектированием, так и различными средствами его автоматизации. В процессе проектирования ПО для выражения его характеристик используются различные нотации – блок-схемы, ER-диаграммы, UML-диаграммы, DFD-диаграммы, а также макеты.

Проектированию обычно подлежат:

Архитектура ПО;

позволяющей описать систему на концептуальном уровне (рис.4.2).

Прецедент – возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Основное назначение диаграммы – описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

- чётко отделить систему от её окружения;
- определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;
- определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть прецедентов).

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ).

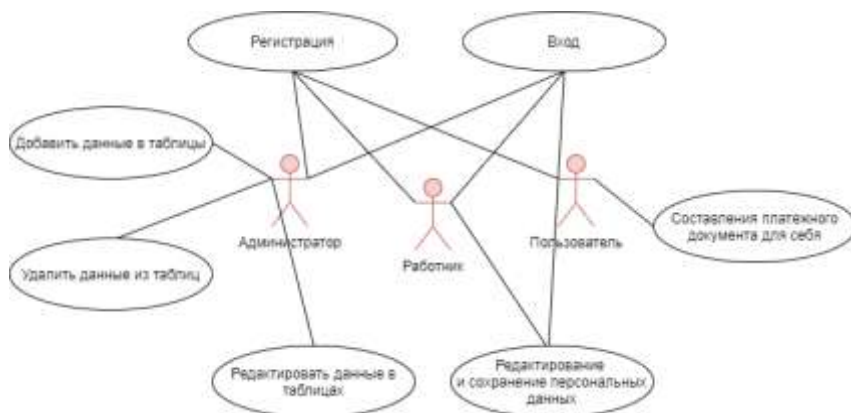


Рис. 4.2 Диаграмма прецедентов

Диаграмма последовательностей служит основным способом расшифровки последовательности действий в процессе выполнения того или иного варианта использования (рис. 4.3).

Таким образом, диаграмма последовательностей всегда создается в привязке к варианту использования. Каждый вариант использования может содержать несколько диаграмм последовательностей, на тот случай, если они описывают несколько альтернативных вариантов развития событий.

Диаграмма последовательностей, так же, как и вариант использования, может быть реализована как в терминах бизнес-объектов, так и в терминах физических сущностей, таких как компоненты или классы.

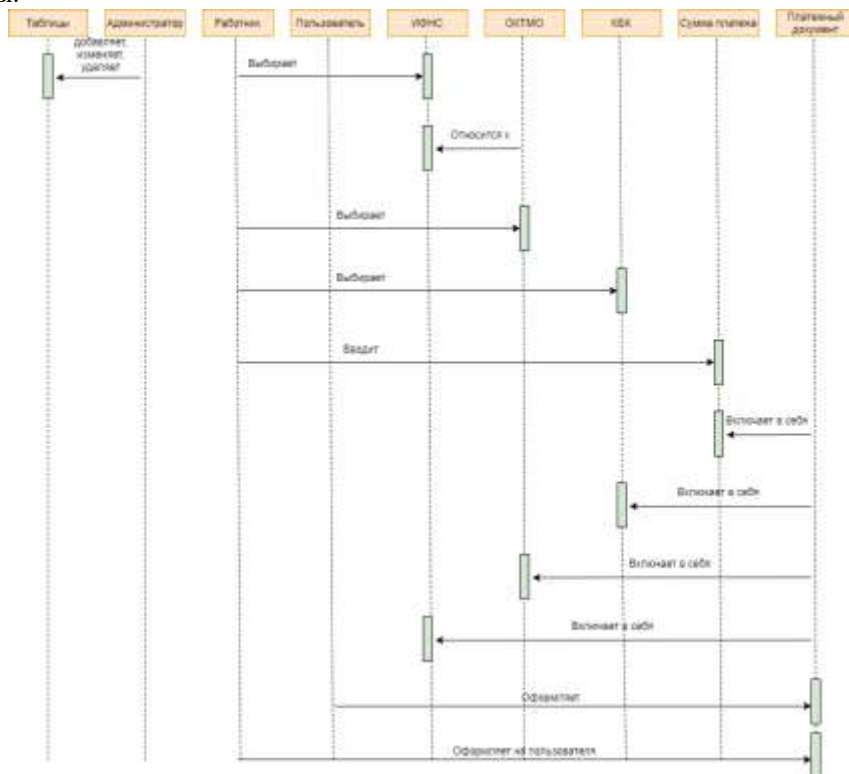


Рис. 4.3. Диаграмма последовательностей

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой «черный ящик» со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня. Наиболее важная функция расположена в верхнем левом углу. А соединяются функции между собой при помощи стрелок и описаний функциональных блоков. При этом каждый вид стрелки или активности имеет собственное значение.

Данная модель позволяет описать все основные виды процессов, как административные, так и организационные (рис. 4.4).

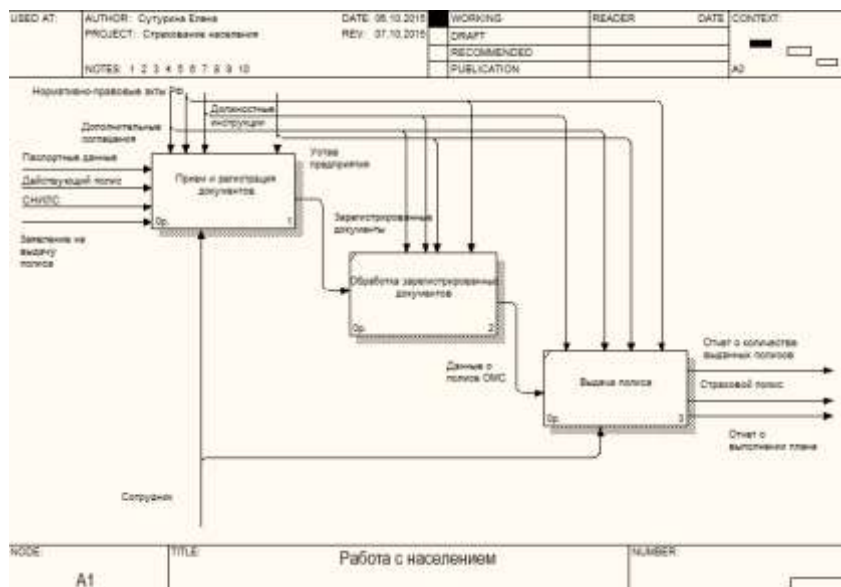


Рис. 4.4. Функциональная модель

Диаграмма деятельности – UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений (рис.4.5).

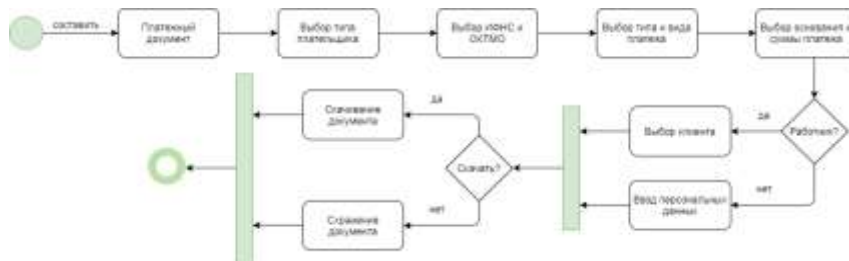


Рис. 4.5. Диаграмма деятельности

Диаграмма классов (рис. 4.6.) определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации).

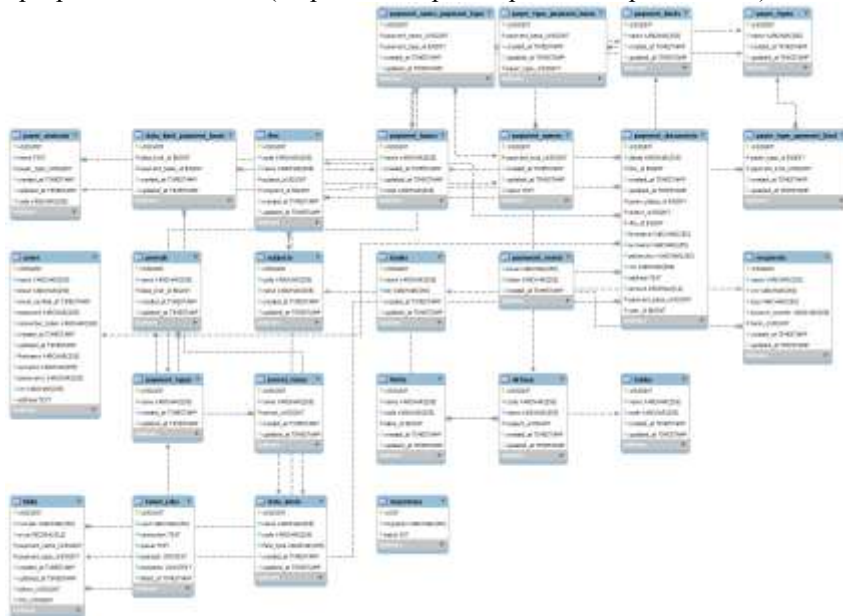


Рис. 4.6. Диаграмма классов

Диаграмма компонентов (рис. 4.7.), в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

Основной тип сущностей на диаграмме компонентов – это сами компоненты, а также интерфейсы, посредством которых указывается

взаимосвязь между компонентами. На диаграмме компонентов применяются следующие отношения:

- реализации между компонентами и интерфейсами (компонент реализует интерфейс);
- зависимости между компонентами и интерфейсами (компонент использует интерфейс).

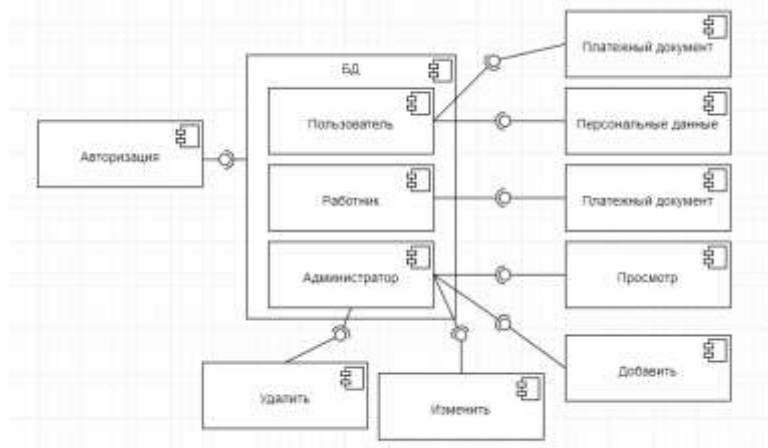


Рис. 4.7. Диаграмма компонентов

3.1.2.2 Обоснование выбора средств разработки

В данном разделе описывается совокупность программных средств, используемых для создания, отладки, функционирования и проверки работоспособности программного продукта.

Обосновывается выбор языка программирования. Приводится перечень проблем, которые ставятся перед разработчиком и описываются возможные пути их решения с помощью выбранного языка программирования. Можно привести сравнительные характеристики различных языков программирования.

Указывается назначение и общие характеристики выбранной среды разработки программы, её возможности, основные области применения. Приводится описание используемых в этой среде средств отладки, рекомендации по их применению.

Обосновывается выбор операционной системы. Указывают наименование, обозначение и характеристику выбранной операционной системы, её версию, в рамках которой будут выполняться разрабатываемые программы. Приводится описание характеристик функций операционной системы, её состав, назначение. Показываются

преимущества выбранной операционной системы по сравнению с другими.

Приводится описание средств, расширяющих возможности операционной системы. Указывают назначение, обозначение и краткую характеристику этих средств, требования к их настройке.

3.1.2.3 Разработка базы данных

При создании базы данных наиболее важными являются задачи, связанные с созданием правильной логической структуры данных, обеспечивающей решение всего набора требуемых задач. Под правильной логической структурой в данном случае понимается структура, созданная с учетом особенностей организации хранения данных, используемых при решении требуемых задач. База данных, разработанная без учета того, как она в дальнейшем будет использоваться, оказывается, как правило, неуклюжей и неэффективной. Создание правильной логической структуры предусматривает комплексный анализ всех факторов, влияющих на формирование и обработку данных.

Цель логического проектирования – преобразование концептуальной модели на основе выбранной модели данных в логическую модель, не зависящую от особенностей используемой в дальнейшем СУБД для физической реализации базы данных. Для ее достижения выполняются следующие процедуры.

1. Выбор модели данных. Чаще всего выбирается реляционная модель данных в связи с наглядностью табличного представления данных и удобства работы с ними.

2. Определение набора таблиц исходя из ER-модели и их документирование. Для каждой сущности ER-модели создается таблица. Имя сущности – имя таблицы. Осуществляется формирование структуры таблиц. Устанавливаются связи между таблицами посредством механизма первичных и внешних ключей. Структуры таблиц и установленные связи между ними документируются.

3. Нормализация таблиц. На этом шаге проверяется корректность структуры таблиц, созданных на предыдущем шаге, посредством применения к ним процедуры нормализации. В результате нормализации получается очень гибкий проект базы данных, позволяющий легко вносить в нее нужные расширения.

4. Определение требований поддержки целостности данных и их документирование. Эти требования представляют собой ограничения, которые вводятся с целью предотвратить помешение в базу данных противоречивых данных. На этом шаге вопросы целостности данных

освещаются безотносительно к конкретным аспектам ее реализации. Должны быть рассмотрены следующие типы ограничений:

- обязательные данные. Выясняется, есть ли атрибуты, которые не могут иметь Null-значений;

- ограничения для значений атрибутов. Определяются допустимые значения для атрибутов;

- целостность сущностей. Она достигается, если первичный ключ сущности не содержит Null-значений;

- ссылочная целостность. Она понимается так, что значение внешнего ключа должно обязательно присутствовать в первичном ключе одной из строк таблицы для родительской сущности;

Сведения обо всех установленных ограничениях целостности данных помещаются в словарь данных.

5. Создание окончательного варианта логической модели данных. На этом шаге подготавливается окончательный вариант ER-модели, представляющей логическую модель данных. Сама модель и обновленная документация, включая словарь данных и реляционную схему связи таблиц, представляется для просмотра и анализа пользователям, которые должны убедиться, что она точно отображает предметную область.

Пример

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап должен быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной системы управления базами данных.

Следуя описанию спроектированной концептуальной модели, можно выделить и описать следующие сущности:

Сущность Subject (Субъект) содержит информацию о субъектах Российской Федерации, является главной по отношению к сущностям Ifns (Ифнс), Oktmo (Октмо) и Payment_document (Платежный документ).

Структура сущности Subject (Субъект):

- id – уникальный идентификатор, первичный ключ;
- code – код субъекта;
- name – название субъекта.

Сущность Payment_document (Платежный документ) содержит информацию о платежных документах. Является подчиненной по отношению к сущностям Payer (Платательщик), Kbk (Кбк), Subject (Субъект).

Структура сущности Payment_document (Платежный документ):


```

$table) {
    $table->id();
    $table->string('code');
    $table->string('name');
    $table->timestamps();
});

```

Свойства полей таблицы Subject (Субъект) представлены в таблице 4.1.

Таблица 4.1 – Характеристики полей таблицы Subject (Субъект).

Имя столбца	Тип данных	Свойства поля
<code>id</code>	<code>id</code>	Ключевое поле: Идентификационное поле: Да (Совпадения не допускаются) <input type="checkbox"/> AUTO_INCREMENT – автозаполнение <input type="checkbox"/> Новые значения: Последовательные <input type="checkbox"/> Размер поля: Длинное целое <input type="checkbox"/>
<code>code</code>	<code>string</code>	Размер поля: 255 <input type="checkbox"/> Подпись: Код субъекта <input type="checkbox"/> Поле не индексированное <input type="checkbox"/> Нет конкретного значения по умолчанию <input type="checkbox"/>
<code>name</code>	<code>string</code>	Размер поля: 255 <input type="checkbox"/> Подпись: Наименование субъекта <input type="checkbox"/> Поле не индексированное <input type="checkbox"/> Нет конкретного значения по умолчанию <input type="checkbox"/>

Эксплуатация начинается с заполнения базы данных реальными данными. На этом этапе необходимо сопровождение базы данных, т.е. проведение контроля непротиворечивости, резервное копирование, архивирование и т.д.

По мере использования базы данных происходит выявление недоработок, уточнение и, возможно, изменение требований к базе данных. В результате может быть принято решение о ее модификации.

3.1.2.4 Логическая и физическая структура сайта

Логическая структура сайта — это система ссылочного взаимодействия между страницами виртуального ресурса. Она должна продумываться на самых ранних этапах разработки сайта и являться основой функциональности дизайна. Логическая структура представлена на рисунке 4.9.

Физическая структура подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой опубликован сайт.

При создании физической структуры следует соблюдать следующие правила:

- нужно назначать имена директорий;
- имена и расширения документов «html» и графических файлов с использованием символов только латинского алфавита и только в строчном регистре;
- стараться, чтобы имена созданных вами файлов и директорий не превышали по длине восьми символов;
- при присвоении имен файлов документам «html» стремиться к тому, чтобы эти имена были «смысловыми»;
- для того чтобы облегчить процесс обновления web-страниц, дополнения разделов или создания новых рубрик;
- рекомендуется создать средство документирования проекта – любую электронную или обычную таблицу, в которую записывайте соответствие элементов физической и логической структуры проекта.

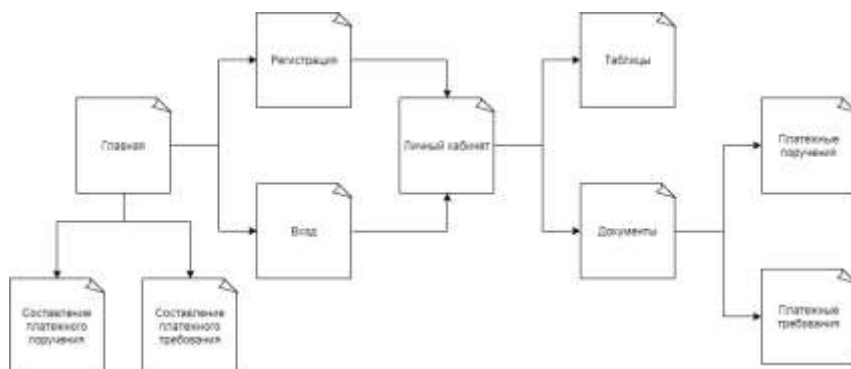


Рис.4.9. Логическая структура сайта

Физическая структура web-приложения:

- структура хранения моделей (рис.);
- структура хранения контроллеров (рис.);
- структура хранения шаблонов (рис.);
- структура хранения скриптов (рис.).

3.1.2.5 Проектирование меню и интерфейса программного продукта

Интерфейс – это граница между объектами системы, через которую они взаимодействуют. Выделяют человеко-машинный интерфейс, как способ общения человека с компьютером (формы, элементы управления, командная строка и др.). Также есть интерфейс

прикладного программирования (API) – описания способов, с помощью которых одно программное приложение может взаимодействовать с другим.

Интерфейс пользователя, он же пользовательский интерфейс (UI – англ. user interface) – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы.

Под совокупностью средств и методов интерфейса пользователя подразумеваются:

- средства: вывода информации из устройства к пользователю и ввода информации/команд пользователем в устройство;
- методы: набор правил, заложенных разработчиком устройства, согласно которым совокупность действий пользователя должна привести к необходимой реакции устройства и выполнения требуемой задачи – так называемый логический интерфейс.

Взаимодействие человека и компьютера в наши дни строится на основе объектно-ориентированного графического интерфейса, в котором:

- оперирование объектами представляется в окнах;
- основным элементом программного управления является меню – выводимый на экран список команд;
- основным элементом аппаратного управления являются манипуляторы.

Основным предназначением приложения является предоставление пользователю необходимой функциональности. Пример интерфейса пользователя представлен на рис. 4.8.

В ходе разработки интерфейса пользователя необходимо определить требования, общие спецификации пользовательских функций и алгоритмов обработки данных.

Требования к графическому интерфейсу:

- Содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки.
- Ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия.
- Удовлетворять правилу "шести" - в одну линейку меню включать не более 6 понятий, каждое из которых содержит не более 6 опций.
- Сохранять стандартизированное назначение и местоположение на экране графических объектов.

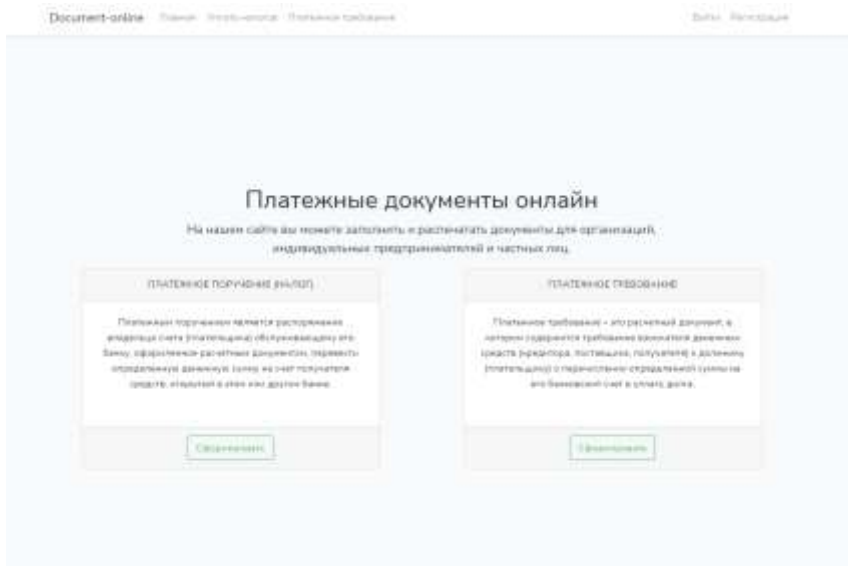


Рис.4.10. Интерфейс веб-приложения по оплате налогов и пошлин

При создании интерфейса, к нему предъявляются следующие требования:

- наличие форм ввода данных;
- наличие форм вывода данных;
- удобная навигация;
- наличие меню;
- наличие всплывающих подсказок;
- защита от некорректного ввода данных;
- обработка исключительных ситуаций;
- наличие сведений о программе;
- наличие сведений об авторе.

3.1.2.6 Описание запросов при разработке программного продукта

Создание запросов в виде хранимых процедур, представлений должно сопровождаться обоснованием их целесообразности и необходимости.

Например, для работы запроса на поиск пациента с определенным талоном был разработан следующий запрос:

Реквизиты платежного документа

Статус лица:*

Основание платежа:*

Налоговый период:

Сумма платежа:*

- Месячные платежи
- Инеретная дата
- Месячные платежи**
- Квартальные платежи
- Полуподровые платежи
- Годовые платежи

Рис. 4.11. Результат выполнения запроса

3.12.7 Проектирование отчетов

Отчет – это форматированное представление данных, выводимое на экран, принтер или файл. Отчет, создаваемый в данной программе, может быть представлен в табличном виде или в свободной форме. Табличные отчеты используются для печати данных, представленных в виде списка.

Пользователь после ввода необходимых реквизитов может получить платежный документ (рис. 4.12).


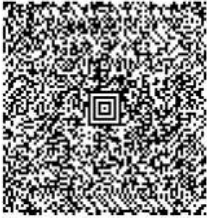
Извещение	Индекс док. 18209965182871512317		(101) 13	Форма №ИД (налог)
				
				
	ФИО Примерный Алгон Петрович		Адрес 198095, ПЛ. СТАЧЕК, Д. 1, Г. САНКТ-ПЕТЕРБУРГ	
	ИНН	Сумма 2000.00		
	Банк получателя СЕВЕРО-ЗАПАДНОЕ ГУ БАНКА РОССИИ	БИК 044030001		
		Сч.№ 0000000000000000000		
	Получатель УФК по г. Санкт-Петербургу (МРИ ФНС России №16 по Санкт-Петербургу)	Сч.№ 40101810200000010001		
		ИНН 7801045990		
		КПП 780101001		
КБК 18210604012021000110		ОКТМО 40303000		
			(107) ГД.00.2018	
			(106) ТП	(110)
Дата		Подпись		
Индекс док. 18209965182871512317		(101) 13	Форма №ИД (налог)	
ФИО Примерный Алгон Петрович		Адрес 198095, ПЛ. СТАЧЕК, Д. 1, Г. САНКТ-ПЕТЕРБУРГ		
ИНН	Сумма 2000.00			
Банк получателя СЕВЕРО-ЗАПАДНОЕ ГУ БАНКА РОССИИ	БИК 044030001			
	Сч.№ 0000000000000000000			
Получатель УФК по г. Санкт-Петербургу (МРИ ФНС России №16 по Санкт-Петербургу)	Сч.№ 40101810200000010001			
	ИНН 7801045990			
	КПП 780101001			
КБК 18210604012021000110		ОКТМО 40303000		
			(107) ГД.00.2018	
			(106) ТП	(110)
Дата		Подпись		
Квитанция				
Отметки банка				

Рис.4.12. Платежный документ

3.1.2.8 Администрирование программного продукта

Администрирование программного обеспечения – это приведение программного обеспечения в соответствие с целями и задачами, для которых оно предназначено. Достигается путём управления, позволяющего минимизировать затраты времени и ресурсов, направляемых на управление системой, и в тоже время максимизировать доступность, производительность и продуктивность системы.

Администрирование базы данных подразумевает под собой разграничение ролей пользователя. Оно необходимо для сохранности и конфиденциальности информации.

Рассмотреть администрирование программного обеспечения, описать категории пользователей, доступный им функционал приложения, а также накладываемые ограничения (рис.4.13).

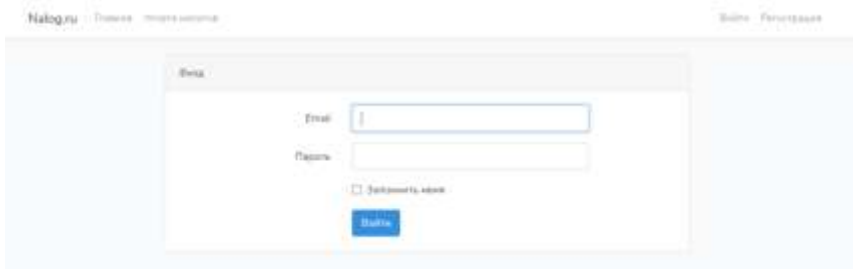


Рис.4.13. Окно авторизации пользователей

3.1.3 Особенности построения части «Описание работы программного продукта»

Данный раздел содержит:

- руководство пользователя;
- руководство программисту.

3.1.3.1 Руководство пользователя

Одним из важных эксплуатационных документов является *Руководство пользователя*. При разработке этого документа следует учитывать следующие рекомендации:

- руководство должно содержать все инструкции, необходимые пользователю;
- изложение должно быть ясным, короткими предложениями;
- следует избегать технического жаргона и узкоспециальной терминологии;
- будьте точны и рациональны – длинные и запутанные руководства обычно никто не читает, например, лучше привести рисунок формы, чем долго ее описывать.

Руководство пользователя содержит следующие разделы:

- общие сведения о программном продукте (наименование программного продукта, краткое описание его функций, реализованных методов и возможных областей применения);
- описание запуска (описание действий по запуску программы и сообщений, которые при этом могут быть получены);
- инструкции по работе или описание пользовательского интерфейса (описание режимов работы, форматов ввода-вывода информации и возможных настроек);
- сообщения пользователю (содержит перечень возможных сообщений, описание их содержания и действий, которые необходимо предпринять по этим сообщениям).

3.1.3.2 *Руководство программисту*

Руководство программиста относится к эксплуатационно-технической документации и требуется в тех случаях, когда система тем или иным образом предоставляет возможность написания, редактирования или использования программного кода.

Примерами могут служить:

- библиотека функций;
- платформа или среда для разработки ПО;
- ПО с открытым кодом.

Документ должен предоставлять всю необходимую информацию для того, чтобы разработчик мог воспользоваться возможностями системы. Для решения этой задачи содержание документа может включать в себя:

- назначение, структуру входных и выходных данных программных функций;
- возможности по созданию программного кода, особенности его интерпретации и компиляции;
- синтаксические особенности используемого языка программирования;
- возможные правила и ограничения при работе с программным кодом;
- различные инструкции по работе с программой.

Список возможных тем этим не ограничивается, все зависит от особенностей конкретной системы. Надо сказать, что руководство программиста бывает очень полезно и для разработчиков системы, являясь справочником по текущей реализации логики работы ПО.

Состав типового руководства программиста

В соответствии с требованиями ГОСТ руководство программиста должно содержать следующие разделы:

– **Назначение и условия применения программы**, где указывают область применения ПО и технические требования, необходимые для его работы.

– **Характеристика программы**, где описывают режим работы программы, показатели скорости ее работы и другие важные для использования характеристики.

– **Обращение к программе**, где указывают способы и параметры запуска программы;

– **Входные и выходные данные**, где описывают формат, способ организации и другие требования к входным и выходным данным;

– **Сообщения**, где приводят тексты сообщений, выдаваемых программой в различных ситуациях и действия, которые необходимо при этом предпринять.

Различные примеры, иллюстрации и таблицы целесообразно

приводить в приложениях к документу.

Стандарты для руководства программиста

ГОСТы регламентируют и этот документ, в данном случае это ГОСТ 19.504. В соответствии с ним определяется структура и содержание *Руководства программиста*.

3.1.4 Особенности построения раздела *Информационная безопасность*

В данном разделе планируется рассмотреть:

- виды возможных угроз программного продукта;
- информационная безопасность программного продукта.

3.1.4.1 Виды возможных угроз программного продукта

В данном разделе основное внимание необходимо уделить вопросам определения возможных потенциальных угроз и действий нарушителей для информации в информационных системах, их классификации и основным методам защиты.

3.1.4.2 Информационная безопасность программного продукта

Программные продукты следует защищать от несанкционированных воздействий различных объектов: человека, технических средств, специализированных программ, окружающей среды. Влияние на программный продукт возможно через применение хищения или физического уничтожения документации на программу или самого машинного носителя, а также путем нарушения работоспособности программных средств.

Программные продукты являются важными объектами защиты по целому ряду причин:

1) они представляют собой продукт интеллектуального труда специалистов высокой квалификации, или даже групп из нескольких десятков или даже сотен человек;

2) проектирование этих продуктов связано с потреблением значительных материальных и трудовых ресурсов и основано на применении дорогостоящего компьютерного оборудования и наукоемких технологий;

3) для восстановления нарушенного программного обеспечения необходимы значительные трудозатраты, а применение простого вычислительного оборудования чревато негативными результатами для организаций или физических лиц.

Защита программных продуктов преследует следующие цели:

- ограничение несанкционированного доступа отдельных категорий пользователей к работе с ними;
- исключение преднамеренной порчи программ с целью нарушения нормального хода обработки данных;
- недопущение преднамеренной модификации программы с целью порчи репутации производителя программной продукции;

- препятствование несанкционированному тиражированию (копированию) программ;
- исключение несанкционированного изучения содержания, структуры и механизма работы программы.

В данном разделе необходимо рассмотреть вопросы информационной безопасности при работе созданного программного продукта.

3.1.5 Особенности построения части «Управление проектом».

В данном разделе приводятся основные понятия проекта, описывается необходимость управления им, применение технологии SMART при управлении проектом. Рассматриваются заинтересованные стороны проекта, устав и планирование проекта, управление стоимостью и рисками проекта.

Проект – это уникальный процесс, состоящий из набора взаимосвязанных и контролируемых работ с датами начала и окончания и предпринятый, чтобы достичь цели соответствии конкретным требованиям, включая ограничения по времени, затратам и ресурсам.

Управление проектом более не используется только в каких-либо исключительных случаях, а, наоборот, все чаще и быстрее становится стандартным способом ведения бизнеса.

Управление проектами — это, говоря обобщенно, комплекс мероприятий, направленных на достижение поставленной цели. Но, когда перед командой стоит задача в сжатые сроки и при ограниченном бюджете создать корпоративный портал или мобильное приложение, уже не до обобщенных формулировок. Чтобы быстро сделать и доставить потребителю качественный работающий продукт, нужны конкретные принципы и методы управления проектами.

Для эффективного управления проект должен быть хорошо структурирован. Суть структуризации сводится к разбивке проекта и системы его управления на подсистемы и компоненты, которыми можно управлять.

Технология SMART — современный подход к постановке работающих целей. Система постановки smart — целей позволяет на этапе целеполагания обобщить всю имеющуюся информацию, установить приемлемые сроки работы, определить достаточность ресурсов, предоставить всем участникам процесса ясные, точные, конкретные задачи.

SMART-цель должна быть:

- S – Specific – конкретной;
- M – Measurable – измеримой;
- A – Achievable – достижимой;
- R – Relevant – значимой;
- T – Time bound – ограниченной во времени.

Цель любого проекта — создание уникального продукта, не имеющего аналогов. Так что можно смело сказать, что проектный подход — это основа внедрения инноваций. С помощью грамотного управления проектами компания может повысить свою эффективность.

Пример:

Цель дипломного проекта по SMART:

1. создать сайт «Hobby-shop» для продажи творческих готовых решений как подарок;
2. сайт отображает категории товаров и товары по этой категории, можно добавлять товары в корзину, для пользователей предусмотрен личный кабинет, для администратора – возможность редактировать, добавлять и удалять товары, категории товаров и заказы, а также удалять пользователей и подтверждать заказ;
3. для разработки используется фреймворк Laravel;
4. есть множество людей, которые создают товары ручного труда и нуждаются в площадке для продажи их продукции;
5. проект должен быть готов в срок до 16 мая 2022 года.

Стейкхолдеры (или заинтересованные лица) – это группы людей или отдельные люди, которых проект как-то затрагивает (как в хорошем, так и в плохом смысле) либо те, кого проект не затрагивает, но они сами могут его «затронуть» или как-то на него повлиять, используя имеющиеся у них возможности или полномочия, пример заинтересованных сторон приведен в табл.4.2.

Заинтересованная сторона может быть как внешней, так и внутренней по отношению к проекту.

Информация об ожиданиях заинтересованных сторон может в первую очередь повлиять на такие аспекты управления проектом, как сбор и анализ требований к результатам проекта и анализ рисков проекта.

Реестр заинтересованных сторон – документ, содержащий описание всех заинтересованных сторон проекта. Заинтересованной стороной в данном случае считается физическое или юридическое лицо, которое каким-либо образом может влиять на ход реализации проекта

Таблица 4.2 – Реестр заинтересованных сторон

Стейкхолдеры	Внутренний / внешний	Что ему нужно?	Что нам нужно?	Риск неосведомленности
Авторы уникальных работ	Внешний	Регулярные заказы, денежная плата	Возможность стабильной покупки готовых работ	Невозможность вести бизнес, репутационный ущерб
Покупатели уникальных работ	Внешний	Покупка уникальных работ	Желание приобрести уникальные работы «Hand made»	Репутационный ущерб
Государство	Внешний	Уплата налогов	Уменьшение налоговой нагрузки	Уголовная ответственность
Местные власти	Внешний	Соблюдение законов	Отсутствие задержек при выдаче документов	Уголовная ответственность
Конкуренты	Внешний	Доля рынка	Честная конкуренция	Репутационный ущерб
Мазнина Юлия Александровна, преподаватель курса «Основы управления проектом», консультант	Внешний	Успешная защита проекта	Предоставление теоретического материала и проверка	Подача неверной информации вышестоящему руководству
Я (студент, руководитель проекта)	Внутренний	Успешная защита проекта	Успешно защитить проект	Репутационный ущерб
Чернега Вадим Александрович	Внешний	Успешная защита проекта	Предоставление теоретического материала и проверка	Подача неверной информации
Сидорова Наталья Викторовна	Внешний	Успешная защита проекта	Успешно защитить проект	Дипломный проект не может быть защищен

Устав проекта - основополагающий документ проекта. Разработка Устава проекта - процесс, который относится к группе процессов инициации проекта. Он предусматривает предварительное разграничение ролей и обязанностей, описывает цели проекта, определяет основные заинтересованные стороны и определяет полномочия менеджера проекта.

Менеджер осуществляет сбор дополнительной информации, совместно с куратором организует предварительные совещания с основными участниками и будущими членами проектной группы. В результате данных мероприятий менеджер проясняет связь со стратегией, интересы и ожидания заинтересованных сторон. Становятся понятны потребности, опасения участников, формируется видение продукта, основных ограничений и критериев успеха. Все это вносится в текст устава. Ниже размещен пример формы устава.

Устав проекта документирует бизнес – потребности, текущее понимание потребностей заказчика, а также новый продукт, услугу или результат, который планируется создать, например:

- назначение или обоснование проекта;
- измеримые цели проекта и соответствующие критерии успеха;
- требования высокого уровня;
- описание проекта высокого уровня;
- риски высокого уровня;
- сводное расписание контрольных событий;
- сводный бюджет;
- требования к одобрению проекта (что составляет успех проекта, кто решает, что проект оказался успешным, и кто подписывает проект);
- назначенный менеджер проекта, уровень ответственности и полномочий;
- имя и полномочия спонсора или другого лица (лиц), утверждающего Устав проекта.

Структуру устава проекта хорошо использовать как справочник и чек-лист для оценки возможности успешно завершить проект. А вопросы, которые будут возникать в ходе разработки устава, позволяют на раннем этапе выявить и устранить конфликты и даже своевременно принять решение о закрытии проекта, если станет очевидным, что цели проекта недостижимы в текущих условиях.

По завершении проекта Устав проекта и все изменения к нему включаются в Архив проекта. После завершения проекта, которое фиксируется приказом, изменения в Устав проекта запрещены.

Планирование проекта – непрерывный процесс определения наилучшего способа действий для достижения поставленных целей

проекта с учетом складывающейся обстановки. Планирование является наиболее важным процессом управления проектом, определяющим во времени всю деятельность по осуществлению проекта.

Главная цель планирования – это построение модели реализации проекта, необходимой для координации действий причастных к проекту лиц. Благодаря этой модели устанавливается порядок, согласно которому будут проводиться работы и т.д. Цель планирования состоит в построении модели реализации проекта. Основным результатом стадии планирования является сводный план осуществления проекта, объединяющий результаты планирования по всем функциям управления проектом. Этот документ является главным и определяющим при осуществлении проекта, он выполняет роль модели (плана) действий и прогноза состояния осуществления проекта и его окружения (рисунок 4.14).



Рисунок 4.14 Дорожная карта проекта

Принципы проектного планирования.

Принимаемые решения и предпринимаемые действия в сфере проектного планирования основываются на нескольких важных принципах:

1. принцип целенаправленности. Выражается в том, что проект направляется на достижение конечной цели инициатора проекта (человека, группы людей, организации и т.д.);
2. принцип системности. Предполагает, что проект управляется как единое целое со своими особенностями формирования и развития, но в то же время может быть разбит на подсистемы с

последующим их изучением, т.к. все они взаимосвязаны и воздействуют друг на друга и на весь проект. Это позволяет найти и создать полезные связи подсистем и их эффективные соотношения, представить качественные и количественные оценки процесса реализации всего проекта и его отдельных элементов;

3. принцип комплексности. Согласно ему, явления рассматриваются с учетом их зависимости и связи, применяются разные методы и формы управления, рассматривается вся совокупность целей проект-менеджмента на различных уровнях и в различных звеньях, отдельные элементы увязываются между собой и соотносятся с основной целью проекта;
4. принцип обеспеченности. Означает, что все предусматриваемые проектом мероприятия должны быть укомплектованы всеми требующимися для их реализации ресурсами;
5. принцип приоритетности. Говорит о том, что при разработке проекта и его реализации основное внимание должно уделяться первостепенным задачам, обусловленным общей концепцией стратегического развития;
6. принцип экономической безопасности планируемых мероприятий. Экономическую безопасность следует рассчитывать, беря за основу вероятность возникновения потерь и убытков как итога неосуществления события, намечавшегося проектом. Никакие нововведения в работе не могут исключать риска, по причине чего в практике разработки и планирования проекта нужно не избегать рисков, а сознательно идти на оправданные риски с целью их снижения до максимально возможного уровня.

В процессе осуществления проекта могут происходить изменения как внутри проекта, так и вне него. Поэтому основное назначение планирования заключается в непрерывном поддержании курса осуществления проекта на пути к его успешному завершению.

Объектами планирования в проекте являются:

1. предметная область;
2. время;
3. стоимость;
4. качество;
5. организация;
6. коммуникации;
7. риски;
8. изменения;
9. прочие компоненты проекта;
10. интеграционный план.

Для целей выполнения дипломного проекта, планирование осуществлялось с помощью построения WBS, диаграммы Ганта.

WBS проекта (она же Work Breakdown Structure или ИСР, Иерархическая Структура Работ) – это разбиение проекта на конкретные результаты, которые должны быть достигнуты для достижения целей проекта.

WBS основана на графической природе, которая помогает менеджерам проектов предсказать результаты, основанные на различных сценариях. Процесс часто описывается как структура ответвления, которая охватывает все этапы проекта в организованном порядке. WBS также может быть представлена в виде табличного списка задач и элементов в плане разбивки работ диаграмм Ганта.

Менеджеры используют структуру декомпозиции, чтобы структурировать и делить проекты на легко управляемые компоненты. Они, в свою очередь, разделяются до тех пор, пока они не назначаются конкретному специалисту в команде.

Почему стоит использовать WBS:

1. помогает правильно организовать проекты;
2. оказывает помощь в описании содержания проекта для заинтересованных сторон;
3. помогает распределить обязанности;
4. показывает основные этапы проекта и все ракурсы контроля;
5. позволяет правильно оценить затраты, риски и время работ.

Компоненты WBS

Согласно иерархической структуре WBS, необходимо пройти несколько этапов (компонентов) для того, чтобы оптимизировать и упростить процесс управления проектом:

1. Утверждение концепции проекта;
2. этапы, которые зависят от размера проекта;
3. задачи с промежуточными результатами.

Эти компоненты могут иметь разные названия в зависимости от компании.

Схема WBS для дипломного проекта предоставлена на рис. 4.13

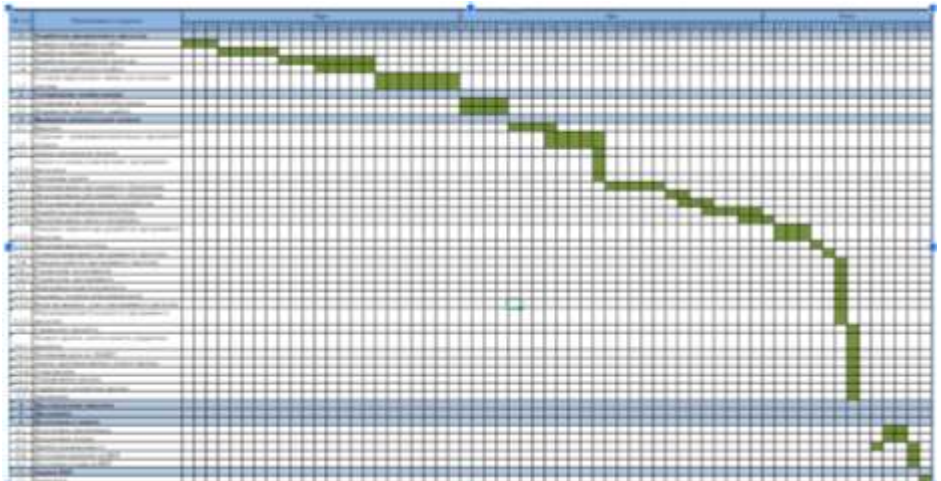


Рисунок 5.2 – Диаграмма Ганта

Управление стоимостью проекта включает в себя процессы, обеспечивающие исполнение и завершение проекта в рамках утвержденного бюджета. Основными процессами при этом являются:

- планирование ресурсов – определение ресурсов (людей, оборудования, материалов) и их количеств, необходимых для выполнения работ проекта;
- оценка стоимости – оценка стоимости ресурсов, необходимых для выполнения работ проекта;
- Разработка бюджета – приложение стоимостных оценок к отдельным работам проекта.
- Управление стоимостью – управление изменениями бюджета проекта.

Трудоемкость разработки программного обеспечения в чел. час определяется по формуле (1).

$$T = T_o + T_i + T_a + T_p + T_{огл} + T_d, \quad (1)$$

где T – общие затраты труда, чел. час;
 T_o – затраты труда на описание задачи;
 T_i – затраты на исследование предметной области;
 T_a – затраты на разработку алгоритма решения задачи;
 T_p – затраты на программирование;

$T_{отл}$ – затраты на отладку программы;
 $T_{д}$ – затраты на подготовку документации;
 Все составляющие определяются через условное число операторов Q (2).

$$Q = q * c * 1 + p, \quad (2)$$

где q – число операторов;
 c – коэффициент сложности задачи (принимается от 1,25 ... 2);
 p – коэффициент коррекции программы, учитывающий новизну проекта (для совершенно новой программы равен 0,1).

$$Q = 950 * 1,6 * (1 + 0,4) = 2128$$

T_o – принимаем в диапазоне от 30 до 40 чел. час (в зависимости от сложности продукта)

T_i определяется по формуле (3).

$$T_i = Q * B_{Si} * k, \quad (3)$$

где B – коэффициент увеличения затрат труда в следствие недостаточного описания задачи (1,2 – 1,5);

S_i – количество операторов, приходящихся на 1 чел. час (75-85);

k – коэффициент квалификации работника.

$$T_i = (2128 * 1,3) / (80 * 0,8) = 43,23 \text{ чел/час}$$

T_a рассчитывается по формуле (4).

$$T_a = Q S_a * k, \quad (4)$$

где S_a – количество операторов на разработку алгоритма решения задачи, приходящихся на 1 чел. час (20 – 25).

$$T_a = 2128 / (23 * 0,8) = 115,65 \text{ чел/час}$$

Затраты труда на программирование найдем по формуле (5).

$$T_{п} = Q S_{п} * k, \quad (5)$$

где $S_{п}$ – количество операторов на программирование, приходящихся на 1 чел. час (20-25).

$$T_{п} = 2128 / (22 * 0,8) = 120,9 \text{ чел/час}$$

Затраты труда на отладку программы находим по формуле (6).

$$T_{отл} = Q S_{отл} * k_{20}, \quad (6)$$

где $S_{отл}$ – количество операторов на отладку программы, приходящихся на 1 чел. час (4-5).

$$T_{отл} = (2128 / (4 * 0,8)) / 20 = 33,25 \text{ чел./час}$$

Затраты на подготовку документации по задаче определяются по формуле (7).

$$T_{д} = T_{др} + T_{до}, \quad (7)$$

где $T_{др}$ – затраты труда на подготовку материала рукописи;

$T_{до}$ – затраты труда на редактирование, печать и оформление документов.

Затраты труда на подготовку материала рукописи определяются по формуле (8):

$$T_{др} = Q S_{др} * k_{10}, \quad (8)$$

где $S_{др}$ - количество операторов на подготовку материала рукописи, приходящихся на 1 чел. час (15 – 20).

$$T_{др} = (2128 / (16 * 0,8)) / 10 = 16,62 \text{ чел./час.}$$

$T_{до}$ определяется по формуле (9):

$$T_{до} = 0,75 * T_{др}, \quad (9)$$

$$T_{до} = 0,75 * 16,62 = 12,46 \text{ чел. час.}$$

$$T_{д} = 16,62 + 12,46 = 29,08 \text{ чел. час.}$$

$$T = 33 + 43,23 + 115,65 + 120,9 + 33,25 + 29,08 = 375,11 \text{ чел. Час.}$$

Полученное значение общей трудоемкости необходимо скорректировать с учетом уровня языка программирования, используя формулу (10):

$$T_{кор} = T * k_{кор}, \quad (10)$$

где $k_{кор}$ - коэффициент учитывающий уровень языка программирования (0,8 – 1).

$$T_{кор} = 375,11 * 0,85 = 318,84$$

Согласно статье 129 Трудового Кодекса России заработная плата (оплата труда работника) – это вознаграждение за труд в зависимости от квалификации работника, сложности, количества, качества и условий выполняемой работы, а также компенсационные выплаты (доплаты и

надбавки компенсационного характера, в том числе за работу в условиях, отклоняющихся от нормальных, работу в особых климатических условиях и на территориях, подвергшихся радиоактивному загрязнению, и иные выплаты компенсационного характера) и стимулирующие выплаты (доплаты и надбавки стимулирующего характера, премии и иные поощрительные выплаты).

Часовая тарифная ставка инженера технолога определяется по формуле (11):

$$T_{ст} = MPOТBф, \quad (11)$$

где МРОТ – минимальный размер оплаты труда с 1 января 2021 г. (оклад сотрудника). В Челябинской области с 1 января 2022 года при расчете выплат работникам применяют новый российский минимальный размер оплаты труда в размере 13890 рублей;

Вф – фактически отработанное время (168 час. в месяц т.е. 21 смена *8 часов).

$$T_{ст} = 13890 / 168 = 82,68 \text{ руб.}$$

Время работы персонального компьютера при создании программного продукта определяется по формуле
Время работы персонального компьютера при создании программного продукта определяется по формуле (12)

$$Фв = 1,15 * Tд + Tп + Tотл * kкор, \quad (12)$$

$$Фв = 1,15 * (29,08 + 120,9 + 33,25) * 0,85 = 179,1 \text{ час}$$

Себестоимость продукции (услуги). – это сумма затрат на производство и реализацию (сбыт) этой продукции (услуг).

Затраты на производство продукции – это расходы по обычным видам деятельности, которые связаны с производством продукции, выполнением работ, оказанием услуг. В совокупности с расходами на продажу они формируют затраты на производство и реализацию продукцию.

Таблица 4.3 – Расчёт заработной платы за проект

Наименование статьи	Формула для расчета	Сумма затрат
1	2	3
основная зп с учетом коэффициента корректировки	$ЗПо = (Тотл + Тп) * k_{кор} * Тст + (Тдр + Та) k_{кор} * Тст + (Тп + Ти + То) k_{кор} * Тст = (33,25 + 120,9) * 0,85 * 82,68 + (16,62 + 115,65) * 0,85 * 82,68 = 33982,92 \text{ руб.}$	33 982,92 руб
дополнительная заработная плата	$ЗПд = 0,2 * ЗПо = 0,2 * 33982,92 = 6796.58 \text{ руб.}$	6 796.58 руб.
отчисления на социальные нужды	$ЕСН = (ЗПо + ЗПд) * 0,3 = (33982,92 + 6796.58) * 0,3 = 12233,85 \text{ руб.}$	12 233,85 руб.
расходы на материалы и запчасти	Хостинг timeweb 40 рублей	40 руб.
затраты	2 % от балансовой стоимости оборудования стоимость ПК $40000 \text{ руб} * 0,02 = 800 \text{ руб}$	1000 руб.
затраты на потребляемые энергоресурсы	$Зэ = 0,40 \text{ кВт}\cdot\text{ч} * 3,74 \text{ руб.} * 180 \text{ ч.} = 2369,28 \text{ руб.}$	263,28 руб.
итого (зи):	$C = 33982,92 + 6796.58 + 12233,85 + 40 + 1000 + 263,28 = 44 566,6$	54 316,63 руб..

В итоге себестоимость проекта равна 54 316,63 руб.

Управление рисками – это систематический процесс снижения неопределенности и управления вероятностью событий в проекте. Цель управления рисками проекта – повышение вероятности возникновения и воздействия благоприятных событий и снижение вероятности возникновения и воздействия неблагоприятных для проекта событий.

К общим проектным рискам обычно относятся:

1. риски в виде отсутствия поддержки со стороны руководства заказчика;

2. отсутствие четкого разделения ответственности между проектными командами заказчика и исполнителя, а также внутри самой проектной команды;
3. недостаток времени у проектной команды заказчика в связи с выполнением текущих функциональных обязанностей;
4. отсутствие соответствующей компетенции у персонала;
5. предоставление неполноценной или несогласованной информации;
6. недоступность технической информации для исполнителя и прочее.

В проектном управлении существует такое понятие, как «риск-менеджмент». Но нужно понимать, что риск-менеджмент – это не только составление и просчет рисков проекта на определенных его этапы. Риск-менеджмент – это спокойная, без истерик и суеты, сенсорика и разрешение проектному менеджеру думать о том, что что-то пойдет не так.

Управление рисками проекта можно разделить на 3 части:

1. Идентификация, анализ и оценка рисков проекта;
2. Определение мероприятий по управлению рисками;
3. Контроль и мониторинг рисков проекта.

При определении рисков необходимо исходить не только из методологии определения и расчета рисков, но и из опыта реальных проектов внедрения программ ИС, имеющейся базы знаний, областей деятельности компании, в которой реализуется проект. Требуется участие не только исполнителя, который определит основные существующие риски проекта внедрения, но и заказчика, который заложит риски в виду нюансов и особенностей ведения деятельности компании.

Перед стартом проекта проводится работа по идентификации возможных и существующих рисков. Процесс идентификации представляет собой поиск всех возможных рисков, по итогам которого мы должны найти ответы на вопрос: «*Что может пойти не так?*» (для негативных рисков). Но при поиске ответов на вопрос: «*Что может пойти не по плану?*» можно найти и позитивные риски.

При планировании рисков следует учитывать, что большим количеством рисков одновременно и качественно управлять невозможно, поэтому цель качественного анализа рисков проектов состоит в группировке и расстановке приоритетов.

Оценка рисков проекта и идентификация проводится для составления плана реагирования на риски. Оптимально управлять одновременно не более 10 рисками.

В процессе расчета величин риска определяются стратегии ликвидации рисков. При этом можно использовать следующие мероприятия по реагированию на риски:

Уклонение от риска предполагает корректировку плана управления проектом так, чтобы максимально исключить угрозу негативного риска, а также снизить последствия риска для целей проекта (например, пересмотреть график проекта или изменить объемы путем исключения некритичных модификаций). Важно заметить, что некоторых рисков можно избежать, если на ранних стадиях проекта детализировано собирать требования заказчика, налаживать более приятные коммуникации.

Передача риска осуществляется путем переложения негативных последствий риска под ответственность третьей стороны. При этом необходимо понимать, что при передаче управления риском третьей стороне сам риск не снимается. Такая стратегия управления риском эффективна в части финансовых рисков. Например, страхование, гарантии выполнения контракта, выдача гарантийных обязательств и т.п. Естественно, что условия передачи рисков третьей стороне несут в себе обязательства по выплате премии за риск третьей стороне. Если проект подразумевает оплату фактических издержек заказчиком, то бремя выплаты премии может быть переложено на заказчика, а вот с фиксированной ценой контракта это бремя обычно ложится уже на исполнителя.

Снижение риска предполагает уменьшение вероятности и/или последствий рискованного события до необходимых пределов. То есть стратегия состоит в формировании предупредительных мер по снижению вероятности негативного события или последствий его наступления, т.к. предупреждение всегда эффективнее, чем разрешение последствий свершившегося факта. Как пример снижения риска можно привести работу по планированию человеческих ресурсов как на стороне заказчика (бизнес-эксперты, ключевые пользователи), так и на стороне исполнителя (консультанты, разработчики) в случаях их болезни, отпуска или увольнения. Как предупредительная мера может использоваться оптимизация сложных процессов за счет их упрощения (исключить неактуальные действия или пересмотреть схему движения бизнес-процесса), увеличение количества тестовых испытаний на реальных данных с активным участием пользователей, выполняющих соответствующие функциональные обязанности и прочее.

Использование риска выбирается в качестве стратегии реагирования на благоприятные последствия случившегося события. Примером использования может служить возникновение возможности по привлечению специалиста более высокого уровня с целью сокращения времени на реализацию определенных задач проекта. Также примером является отказ от модификации в пользу использования стандартного функционала системы в виду изменения методологии бизнес-процесса.

Усиление через позитивное воздействие на складывающуюся ситуацию позволит повлиять на условия формирования события в положительном ключе. Один из примеров реагирования на такой положительный риск – это привлечение специалиста со стороны заказчика, находящегося на более высоком уровне принятия решений по проекту, в случае каких-то согласований задач по модификациям.

Карта рисков дипломного проекта представлена в таблице 4.5.

Таблица 4.5 – Карта рисков проекта

Наименование риска	Причина возникновения	Последствия	Мероприятия по борьбе	Ожидаемый результат	Периодичность контроля	Ответственный
потеря выполненной работы	сбой в работе операционной системы, аппаратная неисправность, вирус на ПК, скачан файл с вирусом, антивирусное ПО не установлено, не обновлено	недовольные сотрудники, приостановка рабочего процесса, проект удален полностью/частично, увеличение сроков проекта	работа с git	возобновление работы системы	каждый раз, когда внесли рабочие изменения в проект	разработчик
сбой в работе электроэнергии	перепад напряжения, замыкание, плановое отключение	сорванные сроки, нервный разработчик, недовольный заказчик	звонок в ЖКХ, вызов электрика	электроэнергия поступает без сбоев и задержек	каждые 3-4 часа	разработчик электрик
отсутствие интернета	отсутствие электроэнергии.	приостановка рабочего процесса.	вовремя оплачивать	поиск ответов на возникла	2 раза в месяц	разработчик

Наименование риска	Причина возникновения	Последствия	Мероприятия по борьбе	Ожидаемый результат	Периодичность контроля	Ответственный
			услуги	ющие ответы быстро		
нехватка времени на разработку программного продукта и подготовку документации	загруженность разработчика, возникновение непредвиденных ошибок в программе	сорванные сроки защиты	правильное планирование рабочего времени, договориться об оказании помощи разработчику	правильная декомпозиция проекта	ежедневно	управляющий проектом

Управление проектом помогло грамотно распределить время и правильно выстроить последовательность задач, помимо этого, оценить риски, и избежать их дальнейшего возникновения в проекте.

3.2 В приложения рекомендуется включать материалы иллюстрационного и вспомогательного характера: таблицы и рисунки большого формата, дополнительные расчеты, описания применяемого в работе нестандартного оборудования, распечатки с персонального компьютера, другие материалы и документы конструкторского, технологического и прикладного характера.

4 ПОРЯДОК ЗАЩИТЫ ДИПЛОМНОГО ПРОЕКТА

4.1 Подготовка доклада для защиты

Подготовке доклада (речи) на защите дипломного проекта следует уделить особое внимание. Текст выступления составляется заранее и согласовывается с руководителем дипломного проекта. Доклад рекомендуется не читать по тексту, а рассказывать. Он может быть проиллюстрирован таблицами, схемами, рисунками, диаграммами, графиками и т.д. на презентационном материале. Речь должна быть ясной, грамматически правильной, уверенной. К иллюстрациям необходимо обращаться только тогда, когда это требуется по ходу доклада, избегая бесцельного обращения к ним.

Раздаточный материал должен быть снабжен титульным листом с указанием темы дипломного проекта, фамилии, имени и отчества обучающегося.

В своем выступлении выпускник должен отразить:

- содержание проблемы и актуальность исследования;
- цель и задачи исследования;
- объект и предмет исследования;
- методику своего исследования;
- полученные теоретические и практические результаты исследования;
- выводы и заключение.

Примерная структура доклада при защите дипломного проекта:

1. **ВСТУПЛЕНИЕ** доклада должно быть очень коротким, состоять из одной-двух фраз и определять область, к которой относится тема дипломного проекта.

2. После этого необходимо очень четко и коротко сформулировать цель дипломного проекта, дать **ПОСТАНОВКУ ЗАДАЧИ**. Это сразу определяет круг вопросов, которые могут рассматриваться в проекте, и обеспечивает правильное восприятие представляемых материалов доклада.

3. Абсолютное большинство дипломных проектов не являются пионерскими, они базируются на уже известных знаниях, результатах, имеют некую «основу», с которой и начинается творческая часть работы автора. Именно это надо коротко осветить в докладе (речи) как **СОСТОЯНИЕ ВОПРОСА**. Обычно этот материал представлен в обзорных главах дипломного проекта.

4. **ПУТИ РЕШЕНИЯ ЗАДАЧИ** - один из основных разделов доклада. Здесь необходимо кратко рассмотреть возможные подходы к решению поставленной задачи и более подробно представить выбранный автором дипломного проекта, объяснить, как решалась задача, и обосновать правильность принимаемого решения.

5. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ должны давать полное представление о том, чего достиг автор дипломного проекта, насколько полученные результаты оригинальны и соответствуют поставленным целям. Желательно в докладе (речи) перечислить все полученные результаты, а подробнее остановиться на наиболее важных.

6. В каждом дипломном проекте имеются ДОПОЛНИТЕЛЬНЫЕ РАЗДЕЛЫ (экономика, охрана труда), о которых в докладе желательно коротко упомянуть. Можно очень коротко сказать о полученных в этих разделах результатах или назвать темы, которые там рассматриваются.

7. В ЗАКЛЮЧЕНИИ доклада необходимо кратко изложить результаты работы по каждому разделу дипломного проекта.

Предлагаемая структура доклада на защиту является наиболее общей и может конкретизироваться и изменяться в зависимости от особенностей и содержания дипломного проекта, полученных результатов и представленных демонстрационных материалов.

В докладе должны упоминаться ВСЕ представленные ДЕМОСТРАЦИОННЫЕ МАТЕРИАЛЫ. Плакат, о котором в докладе не сказано ни слова, явно является «лишним». Состав демонстрационных материалов может корректироваться до утверждения дипломного проекта и должен наилучшим образом поддерживать доклад.

4.2 Подготовка презентации на защите

Защита дипломного проекта является завершающим, а поэтому наиболее важным этапом обучения. Это мероприятие состоит из двух этапов: презентация работы (доклад) и Ваши ответы на вопросы, задаваемые членами государственной экзаменационной комиссии (непосредственная защита). От того насколько четко по теме и доступно для восприятия слушателей будет сделан доклад, насколько будут вопросы, задаваемые комиссией понятны. Для этого необходимо иметь сам доклад, таблично-справочный материал для каждого члена экзаменационной комиссии, а также презентационное сопровождение, которое может включать в себя как использование мультимедийного оборудования (проектор, экран), на котором будут прокручиваться слайды, так и любой другой материал (плакаты, макеты или образцы продукции).