Факультет Электронного обучения

Кафедра __ЭО_____

Направление _09.03.03_____

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студент

1. Тема выпускной квалификационной работы: Автоматизация документооборота организации ООО "Техторг»

Утверждена приказом университета № _____ от «___» _____ 2019 г.

2. Срок сдачи студентом законченной работы «2» ноября 2019 г.

3. Исходные данные по выпускной квалификационной работе_____

1. Материалы предпроектного анализа предприятия.
2. Законодательная и нормативная документация по теме ВКР.
3. Научная, техническая и учебная литература по теме работы.

---

4. Содержание разделов выпускной квалификационной работы

Введение

I Аналитическая часть

1.1. Технико-экономическая характеристика предметной области и предприятия. Анализ деятельности «КАК ЕСТЬ»
   1.1.1. Характеристика предприятия и его деятельности
   1.1.2. Организационная структура управления предприятием
   1.1.3. Программная и техническая архитектура ИС предприятия
1.2. Характеристика комплекса задач, задачи и обоснование необходимости автоматизации
   1.2.1. Выбор комплекса задач автоматизации и характеристика существующих бизнес процессов
   1.2.2. Определение места проектируемой задачи в комплексе задач и ее описание
   1.2.3. Обоснование необходимости использования вычислительной техники для решения задачи
   1.2.4. Анализ системы обеспечения информационной безопасности и защиты информации
1.3. Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ»
   1.3.1. Анализ существующих разработок для автоматизации задачи
   1.3.2. Выбор и обоснование стратегии автоматизации задачи
   1.3.3. Выбор и обоснование способа приобретения ИС для автоматизации задачи
1.4. Обоснование проектных решений
   1.4.1. Обоснование проектных решений по информационному обеспечению

1.4.2. Обоснование проектных решений по программному обеспечению
1.4.3. Обоснование проектных решений по техническому обеспечению

II Проектная часть

2.1. Разработка проекта автоматизации
2.1.1. Этапы жизненного цикла проекта автоматизации
2.1.2. Ожидаемые риски на этапах жизненного цикла и их описание
2.1.3. Организационно-правовые и программно-аппаратные средства обеспечения информационной безопасности и защиты информации
2.2. Информационное обеспечение задачи
2.2.1. Информационная модель и её описание
2.2.2. Характеристика нормативно-справочной, входной и оперативной информации
2.2.3. Характеристика результатной информации
2.3. Программное обеспечение задачи
2.3.1. Общие положения (дерево функций и сценарий диалога)
2.3.2. Характеристика базы данных
2.3.3. Структурная схема пакета (дерево вызова программных модулей)
2.3.4. Описание программных модулей
2.4. Контрольный пример реализации проекта и его описание

III Обоснование экономической эффективности проекта

3.1 Выбор и обоснование методики расчёта экономической эффективности

3.2 Расчёт показателей экономической эффективности проекта

Заключение

Приложения

4. Основные вопросы, подлежащие разработке.

В главе 1 необходимо представить обоснование актуальности выбора автоматизируемой задачи, проектных решений по информационному, программному и аппаратному обеспечению, дать ее развернутое описание, отразить взаимосвязь с другими задачами, изложить используемую стратегию автоматизации и способ приобретения информационной системы.

В разделе 1.1. необходимо привести краткое описание компании и таблицу показателей ее деятельности, рисунок организационной структуры и его описание, рисунки программной и технической архитектуры, а также их описание [11].

В разделе 1.2 следует обосновать актуальность выбора автоматизируемой задачи, для этого необходимо провести анализ организационной, программной и технической архитектуры с целью определения перечня задач, которые необходимо автоматизировать и выбора наиболее приоритетной из них. С целью определения информационных потоков выбранной задачи необходимо привести соответствующие IDEF диаграммы. Далее необходимо привести рисунок, отражающий документооборот автоматизируемой задачи, таблицу прагматических характеристик соответствующих документов (периодичность, время на обработку и так далее) и определить комплекс программно-аппаратных средств, обеспечивающих информационную безопасность и защиту информации в рамках решаемой задачи [1,2,4,6].

В разделе 1.3 необходимо провести анализ готовых программных решений автоматизируемой задачи (в виде сравнительной таблицы наиболее интересных решений),

определить стратегию автоматизации и способ приобретения информационной системы [2,11].

В разделе 1.4 необходимо провести анализ и обосновать выбор средств, используемых для автоматизации задачи [1,2,4,5,8,9,10]: технического обеспечения (компьютеры и периферия, сетевое оборудование, офисная техника, дополнительные устройства), программного обеспечения (операционная система, СУБД, среда разработки, другое прикладное ПО), информационного обеспечения (классификаторы, справочники, документы, информационные базы).

В главе 2 необходимо представить проектные решения в соответствии с выбранной моделью жизненного цикла: начиная с анализа рисков на всех этапах разработки системы и закачивая примером ее опытной эксплуатации.

В разделе 2.1 необходимо провести выбор стандарта и модели жизненного цикла, соответствующих автоматизируемой задаче, а также стратегии внедрения проектируемой информационной системы. После этого следует перечислить риски для каждого этапа выбранной модели жизненного цикла и определить пути их снижения, в том числе посредством разработки средств информационной безопасности и защиты информации проектируемой системы [2,3,4,7].

Раздел 2.2 отражает процесс проектирования информационного обеспечения задачи [2,8,9,10] и должен содержать:

- рисунок информационной модели и ее описание;
- таблицу используемых систем кодирования и описание классификаторов;
- описание входных и оперативных документов, файлов и экранных форм;
- описание результатных документов, файлов и экранных форм;
- таблицы входных и результатных показателей, а также их описание.

Раздел 2.3 отражает процесс проектирования программного обеспечения задачи [2, 8, 9, 10] и должен содержать:

- рисунки дерева функций и сценария диалога, а также их описание;
- рисунок ER – модели (количество таблиц в ER модели должно быть равно количеству таблиц в информационной модели), описание структуры записей каждой таблицы;
- рисунок дерева вызова программных модулей, а также таблицу с перечнем и назначением модулей;
- блок-схема основного/расчетного модуля, описание блок-схем алгоритмов основных расчетных модулей.

В разделе 2.4 следует представить экранные формы, демонстрирующие работу системы (не менее 7) и их описание.

В главе 3 приводится методика расчета показателей экономической эффективности и расчеты, сделанные в соответствии с изложенной методикой. Расчетные данные следует представить в виде таблиц и диаграмм, отражающие сравнение базового и предлагаемого вариантов.

Приложение обязательно должно содержать фрагмент листинга программного кода (распечатка на исходном языке программирования отлаженных основных расчетных модулей - около 400 операторов языка высокого уровня или адаптированных программных средств, использованных в работе).

5. Список литературы

1. Аппаратное обеспечение вычислительных систем / Д.В. Денисов, В.А. Артюхин, М. Ф. Седненков; под ред. Д.В. Денисова. – М.: Маркет ДС, 2010 – 184 с. (Университетская серия.)
2. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем - М.: ИУИТ, 2012 - 300 с.
3. ГОСТ Р ИСО/МЭК 12207-99, Руководящий документ, Госстандарт России, Москва, 2004
4. Информационная безопасность и защита информации, Мельников В. П., М.: Академия, 2012, - 336 стр.
5. Кузнецов С. Д. Базы данных. Модели и языки – М.: Бином-Пресс, 2013 г. - 720 с.
6. Методология функционального моделирования IDEF0, Руководящий документ, Госстандарт России
7. Модель процессов MSF, Microsoft 2002
8. О.Л. Голицына, И.И. Попов, Н.В. Максимов Базы данных. Учебное пособие. 3-е изд., перераб, и доп. М.: ФОРУМ, 2012. 400 с.
9. Беленькая М.Н., Малиновский С.Т., Яковенко Н.В. Администрирование в информационных системах. Научно-популярное издание. - М.: Горячая линия - Телеком, 2011. - 300 с.
10. Култыгин О.П. Администрирование баз данных. СУБД MS SQL Server. - М: Московская финансово-промышленная академия, 2012 -232 с. (Университетская серия).
11. Устройство и функционирование информационных систем. Н. З. Емельянова, Т. Л. Партыка, И. И. Попов., Учебное пособие. М.:ФОРУМ: НИЦ ИНФРА-М, 2012. 448 с.

Оглавление

Введение.....8
I Аналитическая часть.....11
1.1 Технико-экономическая характеристика предметной области ООО «Техторг» 11
1.1.1 Характеристика предприятия и его деятельности.....12
1.1.2 Организационная структура управления предприятием.....14
1.1.3 Программная и техническая архитектура ИС предприятия.....17
1.2 Характеристика комплекса задач, задачи и обоснование необходимости автоматизации.....21
1.2.1 Выбор комплекса задач автоматизации и характеристика существующих бизнес процессов.....21
1.2.2 Определение места проектируемой задачи в комплексе задач и ее описание.....25
1.2.3 Обоснования необходимости использования вычислительной техники для решения задачи.....27
1.2.4 Анализ системы обеспечения информационной безопасности и защиты информации.....31
1.3 Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ».....32
1.3.1 Анализ существующих разработок для автоматизации задачи.....32
1.3.2 Выбор и обоснование стратегии автоматизации задачи.....36
1.3.3 Выбор и обоснование способа приобретения ИС для автоматизации комплекса задач.....37
1.4 Обоснование проектных решений.....40
1.4.1 Обоснование проектных решений по информационному обеспечению.....40
1.4.2 Обоснование проектных решений по программному обеспечению.....41
II Проектная часть.....49
2.1 Разработка проекта автоматизации.....49
2.1.1 Этапы Жизненного Цикла проекта автоматизации.....49
2.1.2 Ожидаемые риски на этапах жизненного цикла и их описание.....66
2.1.3 Организационно-правовые и программно-аппаратные средства обеспечения информационной безопасности и защиты информации.....68
2.2 Информационное обеспечение задачи.....70

2.2.1 Информационная модель и её описание.....70
2.2.2 Характеристика нормативно-справочной, входной и оперативной информации.....71
2.2.3 Характеристика результатной информации.....72
2.3 Программное обеспечение задачи.....72
2.3.1 Общие положения (дерево функций и сценарий диалога).....72
2.3.2 Характеристика базы данных.....74
2.3.3 Структурная схема пакета (дерево вызова процедур и программ).....90
III Обоснование экономической эффективности проекта.....101
3.1 Выбор и обоснование методики расчёта экономической эффективности.…101
3.2 Расчёт показателей экономической эффективности проекта.....109
Заключение.....111
Список литературы.....114
ПРИЛОЖЕНИЕ 1 Листинг программного кода.....116
ПРИЛОЖЕНИЕ 2 Формы.....124

Введение

Не для кого не секрет, что одним из обязательных условий развития бизнеса является постоянная оптимизация бизнес процессов. В сфере ИТ особенно популярно внедрять разного рода информационные системы для полной автоматизации деятельности как одного отдела, так и всего предприятия в целом. Но, к сожалению, большинство систем такого класса имеют общий функционал и небольшой довесок в виде дополнительных модулей для определенной специализации компании. В наше время очень часто предприятия сталкиваются с проблемой улучшения управляемости компании: улучшение контроля и ускорение бизнес-процессов, улучшение возможности их отслеживания, оптимизация рабочего времени, экономия трудозатрат, повышение производительности труда и так далее. Единственным способом реализации подобных задач является внедрение автоматизированной системы.

В данном дипломном проекте будет исследоваться деятельность компании ООО «Техторг». Данная компания уже давно на рынке России и занимается продажей различной техники, компьютеров и ПО, а также оказывает услуги в сфере дальнейшего сервиса и ремонта.

В рамках снижения ставки рефинансирования, которая падает уже в течении нескольких месяцев, управляющие компанией акционеры приняли решение о сокращении затрат на малоприбыльные направления и оптимизацию ключевых бизнес процессов.

Целью данной дипломной работы является разработка программной информационной системы автоматизации документооборота компании, способной всегда иметь точнейшие сведения о количестве заявок, их видах, сократить время на создание аналитических отчетов и передачу документов в электронном виде, а также иметь возможность составления отчетности на предприятие и выявление задолженности.

Реализация поставленной цели обусловила необходимость решения следующих задач:

- осуществить поиск информации по тематике дипломной работы и провести ее анализ и исследования;

- охарактеризовать исследуемую организацию и вид ее деятельности;
- дать краткую характеристику организационной структуры и программно-технической архитектуры ИС предприятия;
- провести обоснование необходимости автоматизации и выбрать комплекс задач автоматизации;
- выполнить анализ существующих разработок для автоматизации задачи;
- обосновать проектные решения по информационному и программному обеспечению;
- описать этапы жизненного цикла проекта автоматизации и определить риски, а так же обеспечить информационную безопасность и защиту информации;
- описать информационную модель предприятия;
- провести характеристику нормативно-справочной, входной и оперативной информации и результатной информации;
- описать характеристику базы данных;
- выбрать и обосновать методику расчета экономической эффективности;
- провести расчёт показателей экономической эффективности проекта.

В первой главе данного дипломного проекта будет проведен анализ существующей системы обработки документов, будут выявлены слабые места для их более глубокой автоматизации. В компании ООО «Техторг», на данный момент, внедрена система 1С: Предприятие.

Во второй главе будет выбран способ доработки системы для сокращения человеческих ресурсов, используемых компанией. Также в данной главе формируется проект автоматизации задачи на основе данных полученных в первой главе. Выбирается стандарт жизненного цикла, по которому в дальнейшем будет осуществляться проект, описывается информационная модель проекта, входная и выходная информация, используемая в системе, а также процессы преобразования этой информации, рассматриваются программное и технологическое обеспечение задачи. Так же во второй главе рассматривается структура базы данных, необходимой для проекта и рассматривается реализация задачи на контрольном примере.

В третьей главе выбирается методика расчета экономической эффективности проекта, а также производится сам расчет, который показывает положительный результат, что свидетельствует о достижении поставленных в проекте целей. Рассматриваются затрачиваемые ресурсы до автоматизации и после, а также просчитывается срок окупаемости данного проекта автоматизации.

I Аналитическая часть

1.1 Технико-экономическая характеристика предметной области ООО «Техторг»

Компания с ограниченной ответственностью (ООО) - это бизнес-структура, разрешенная в соответствии с государственными законами, регулирующими положения, касающиеся ООО. Владельцы ООО обычно называются учредителями.

Многие не ограничивают право собственности, то есть любой может быть членом, включая частных лиц, корпорации, иностранцев и иностранных юридических лиц, и даже другие ООО. Однако некоторые организации не могут создавать ООО, в том числе банки и страховые компании.

ООО является более формальным партнерским соглашением, которое требует, чтобы устав организации был представлен государству. ООО гораздо проще настроить, чем корпорацию, и она обеспечивает большую гибкость и защиту.

ООО не платят налоги сами. Вместо этого прибыль и убытки перечислены в личных налоговых декларациях владельца (владельцев). Если обнаружено мошенничество или если компания не выполнила требования законодательства и требований к отчетности, кредиторы могут быть в состоянии преследовать участников.

Предприятие ООО «Техторг» занимается продажей и ремонтом компьютерной и бытовой техники. ООО «Техторг» предоставляет своим покупателям возможность широкого выбора товаров и инструментов для самых различных нужд. Вся продукция в магазинах представлена авторитетными российскими и зарубежными производителями.

Несомненным преимуществом магазинов является оказание квалифицированных консультационных услуг. Квалифицированные консультанты помогут подобрать необходимую продукцию, исходя из конкретных задач. Также в магазинах есть возможность проконсультироваться у опытных ИТ-специалистов, которые научат и объяснят как правильно подобрать компьютер для определенных целей, нужных покупателям, либо проконсультируют по поводу возможного ремонта либо апгрейда уже имеющегося у клиента ПК.

Предприятие «Техторг» представляет на рынке следующие виды продукции:

- компьютеры;
- ноутбуки;
- телевизоры;
- бытовая техника;
- инструменты и оборудование;
- оргтехника;
- автотовары.

Вся выпускаемая продукция соответствует государственным стандартам и заложенным техническим требованиям.

1.1.1 Характеристика предприятия и его деятельности

В компании «Техторг» есть простой девиз: «Наша репутация основана на обслуживании клиентов». Вот почему она делает все возможное, чтобы клиенты получали персональное внимание, квалифицированную помощь и послепродажное обслуживание, которого они заслуживают.

Сеть «Техторг» осуществляет свою деятельность с 2007 года. Более 50 магазинов сети работает в 35 городах России.

За это время магазин понял, что продажа самых лучших брендов - таких как QUMO, Hansa, Smarbuy, IRBIS и Indesit - по низким конкурентоспособным ценам является важной частью успеха. Кроме того, узнали, что хорошая устная реклама - самый ценный актив. Вот почему прилагает все усилия, чтобы обеспечить превосходное качество обслуживания клиентов. Помимо эффективного формата розничной торговли, ориентированной на покупателя концепции магазина, компания предлагает своим клиентам высококлассную сервисную поддержку под брендом «Техторг.Сервис».

Также «Техторг» активно развивает стратегию интегрированных продаж для унификации товарного ассортимента и создания бесшовного покупательского опыта как в обычных гипермаркетах, так и в интернете.

К тому же осуществление ремонта и сервиса в самой компании значительно повышает доверие клиентов к организации. Гораздо проще обращаться в те же структуры, в которых была произведена покупка или же осуществить обращение, в случае если потребовался какой-либо ремонт, чем бегать и искать сервис который примет ваш заказ. Также можно получить квалифицированную консультацию по различным вопросам, которые входят в компетенцию данной организации.

Благодаря всем этим показателям, компания приобрела многолетний опыт, высококлассных специалистов и множество верных клиентов, которые ценят ее деятельность, качество обслуживания и товаров, ежедневно пользуются ее услугами сервиса и ремонта оборудования.

В таблице 1.1 приведены основные экономические показатели последних двух лет работы предприятия и их сравнение. Из таблицы видно, что организация является прибыльной и рентабельной.

Таблица 1.1

Основные экономические показатели ООО «Техторг».

Показатели	**2017 год**	**2018 год**	**Отклонение 2018г. от 2017г**
Выручка от продаж, тыс. руб.	9 237 425	9 517 256	+279 831
Себестоимость проданных товаров, тыс. руб.	7 389 940	7 613 804	+223 864
Прибыль от продаж, тыс. руб.	1 847 485	1 903 451	+55 966
Прибыль от ремонтов, тыс. руб.	257 365	293 521	+36 156
Расходы (налоги, заработная плата, аренда), тыс. руб.	1 212 325	1 253 235	+92 910
Чистая прибыль, тыс. руб.	892 525	943 451	+50 926

1.1.2 Организационная структура управления предприятием

Организационная структура ООО «Техторг» показана на рис. 1.1, в ней выделяются функциональные подразделения, а также подразделение по управлению магазинами сети. Такая структура создана вследствие увеличения размеров предприятий, усложнения технологических процессов в условиях динамически меняющегося окружения. Головной офис осуществляет реализацию всех процессов управления объектами в едином органе управления, который осуществляет сбор и обработку информации об управляемых объектах и на основе их анализа в соответствии с критериями системы вырабатывает управляющие сигналы.

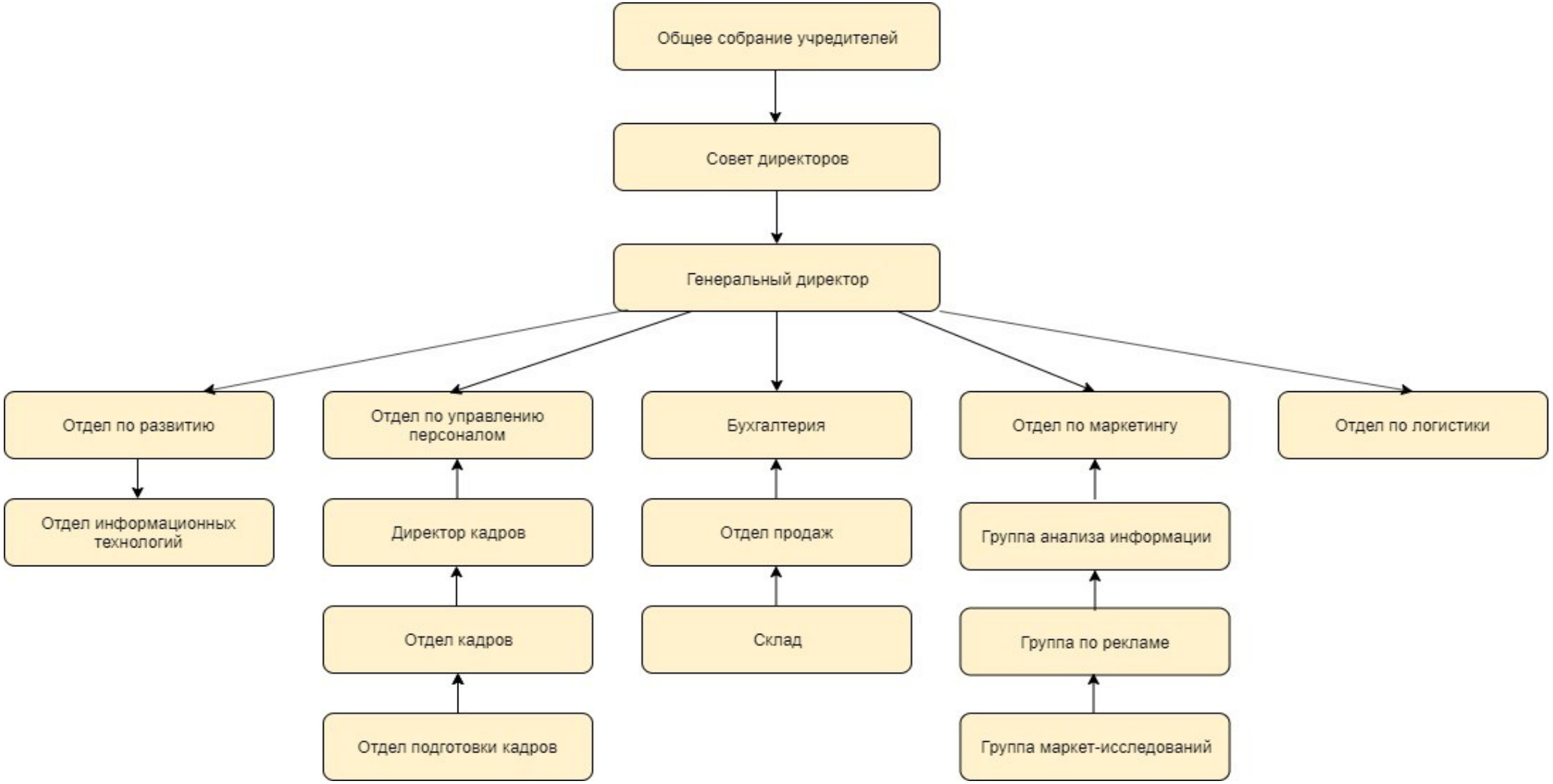


Рисунок 1.1 - Организационная структура ООО «Техторг»

Отдел по развитию – осуществляет поиск новых объектов. Ведение проектов по открытию новых магазинов. Организацию строительно-ремонтных работ и документации.

Отдел информационных технологий – занимается обеспечением правильной эксплуатации и бесперебойной работы, а также технического обслуживания компьютеров, организационной, вычислительной техники, серверного оборудования учреждений и программного обеспечения.

Отдел по управлению персоналом – производит расчет заработной платы. Разработку систем мотивации и премирования персонала.

Директор кадров – выполняет управление кадрами.

Отдел кадров – осуществляет процесс подбора персонала.

Отдел подготовки кадров – обеспечивает развитие и обучение персонала.

Бухгалтерия – выполняет учет товаров, подготовку налоговых отчетов, распределение зарплаты.

Отдел продаж - развитие структуры продаж розницы. Управление сетью розничных магазинов. Обеспечение достижения ключевых показателей: оборот, прибыль, уровень сервиса и т.д.

Склад – осуществляет хранение, прием и отгрузку товаров. Проводит тестирование, распаковку и упаковку продукции.

Отдел маркетинга – управление группами по маркетингу.

Группа анализа информации – изучение мнений потребителей о товаре компании.

Группа по рекламе – планирование, разработка и проведение рекламных мероприятий.

Группа по марке-исследований – разработка маркетинговых мероприятий с целью привлечения новых покупателей в магазины.

Отдел по логистике - координация взаимодействия отделов в цепочке поставок товара. Оптимизация бизнес-процессов логистических служб. Повышение эффективности использования складских мощностей. Взаимодействие с представителями сертификационных и таможенных органов.

1.1.3 Программная и техническая архитектура ИС предприятия

Техническая архитектура «Техторг» представляет собой сервер баз данных (БД), почтового сервера, файлового сервера, принтеров и компьютеров на рабочих местах сотрудников. Рабочие места сотрудников предприятия оснащены необходимым набором оборудования.

Технические характеристики оборудования, используемого на предприятии, показаны в таблице 1.2

Таблица № 1.2

Технические характеристики оборудования, используемого в «Техторг»

№	Наименование оборудования	Характеристики	Цель применения
1	Компьютеры сотрудников	Intel Core i3-6300 3.8 ГГц, RAM 4 Гб, SSD 240 Гб	Ведение документации, создание отчетов.
2	Сетевое оборудование	Asus RT-N66U	Обеспечивает подключение пользователей к интернету
3	Коммутационное оборудование	Asus GigaX 3112	Обеспечивает взаимодействие в локальной сети либо группы локальных сетей
4	Принтер/ сканер/ копир	HP LaserJet 3055	Печать/ копирование документации
5	Сервер баз данных, Файл-сервер, Почтовый сервер	Процессор Intel® Xeon® E5-2643 3.8 ГГц, RAM 32Гб, 10x2 ТБ	Обслуживает БД, создает резервные копии, обслуживает запросы пользователей, используется как хранилище файлов и документов, обеспечивает прием и передачу почтовых сообщений между пользователями

Существующая архитектура ИС, используемая на предприятии, показана на рис.1.2

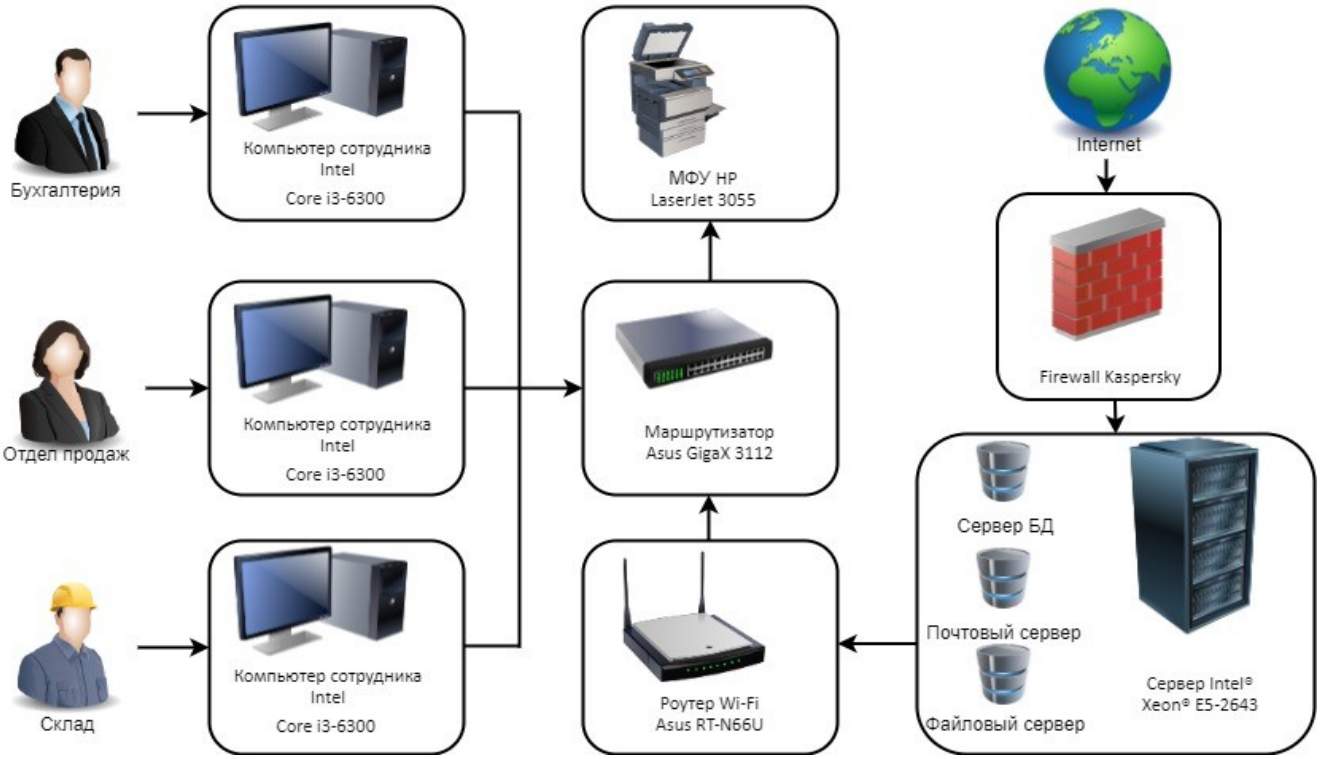


Рисунок 1.2 – Техническая архитектура ООО «Техторг»

Так как электронный документооборот очень мал, используется один главный сервер и три промежуточных базы данных (дополнительных) для обеспечения безопасности, обмена сообщениями для совместной работы и сервер базы 1С: Предприятие, которые своевременно и при необходимости обновляют. Это необходимо для удобства, гибкости и большей безопасности общей системы в целом.

На предприятии на компьютерах сотрудников используется набор офисных приложений (пакет MS Office 2007, Google Chrome web-browser) и ПО 1С: Предприятие, работающих под операционной системой Windows 7.

Имеющаяся ИС для работы использует систему управления базами данных (СУБД) MS SQL Server 2017 установленная на ОС Microsoft Windows Server 2008 R2, которая позволяет создавать резервные копии данных и, в случае необходимости, восстанавливать их.

Для ввода информации о клиентах во внутреннюю базу данных компании используется интерфейс СУБД MS Access 2017 со связью с базой данных в MS SQL Server.

Обеспечение безопасности осуществляется с помощью последних антивирусных приложений фирмы Kaspersky (антивирус, firewall, анти-спам), который дает полную защиту предприятия от виртуальных взломщиков и позволяет сохранять всю нужную информацию для служебного пользования.

Для управления почтовыми сообщениями на предприятии служит MS Exchange Server.

Существующая схема программной архитектуры используемая на предприятии, показана на рис. 1.3

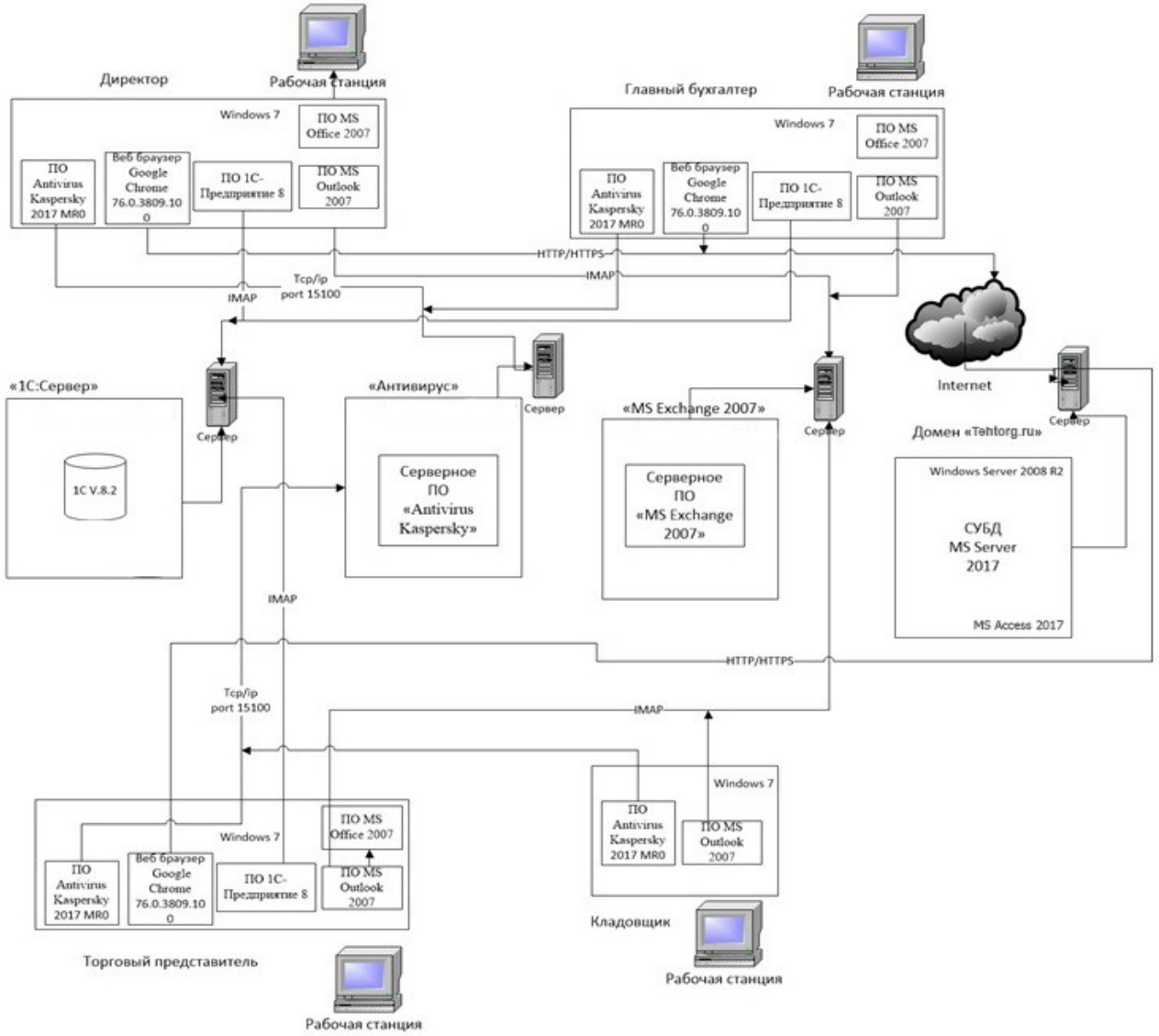


Рисунок 1.3 – Схема программной архитектуры ООО «Техторг»

1.2 Характеристика комплекса задач, задачи и обоснование необходимости автоматизации

1.2.1. Выбор комплекса задач автоматизации и характеристика существующих бизнес процессов

Существующие бизнес-процессы «Техторг» вытекают из набора деловых связей, обозначенных в организационной структуре компании. На рис. 1.4 показан процесс работы компании

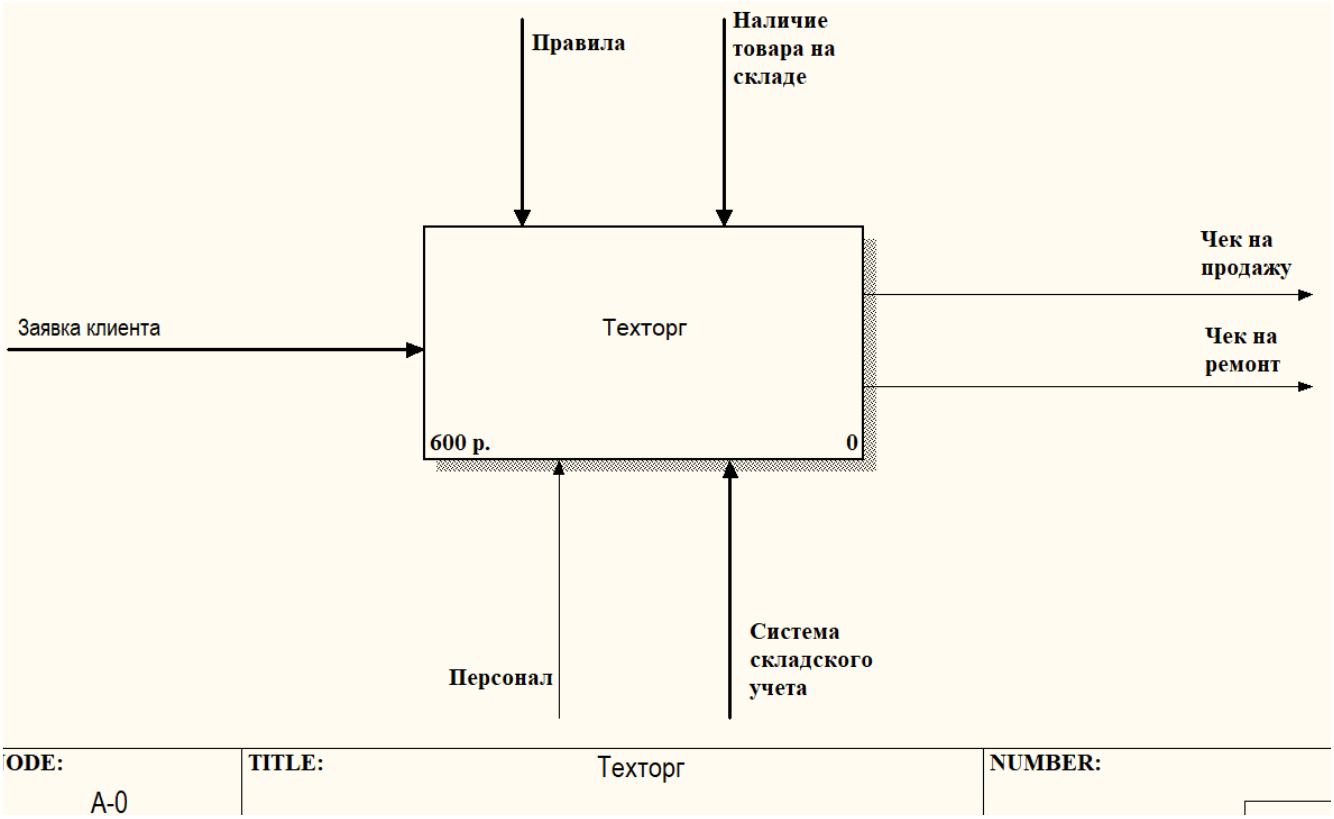


Рисунок 1.4 – IDEF0Техторг A-0

Декомпозиция DEF0 Техторг показана на рис. 1.5

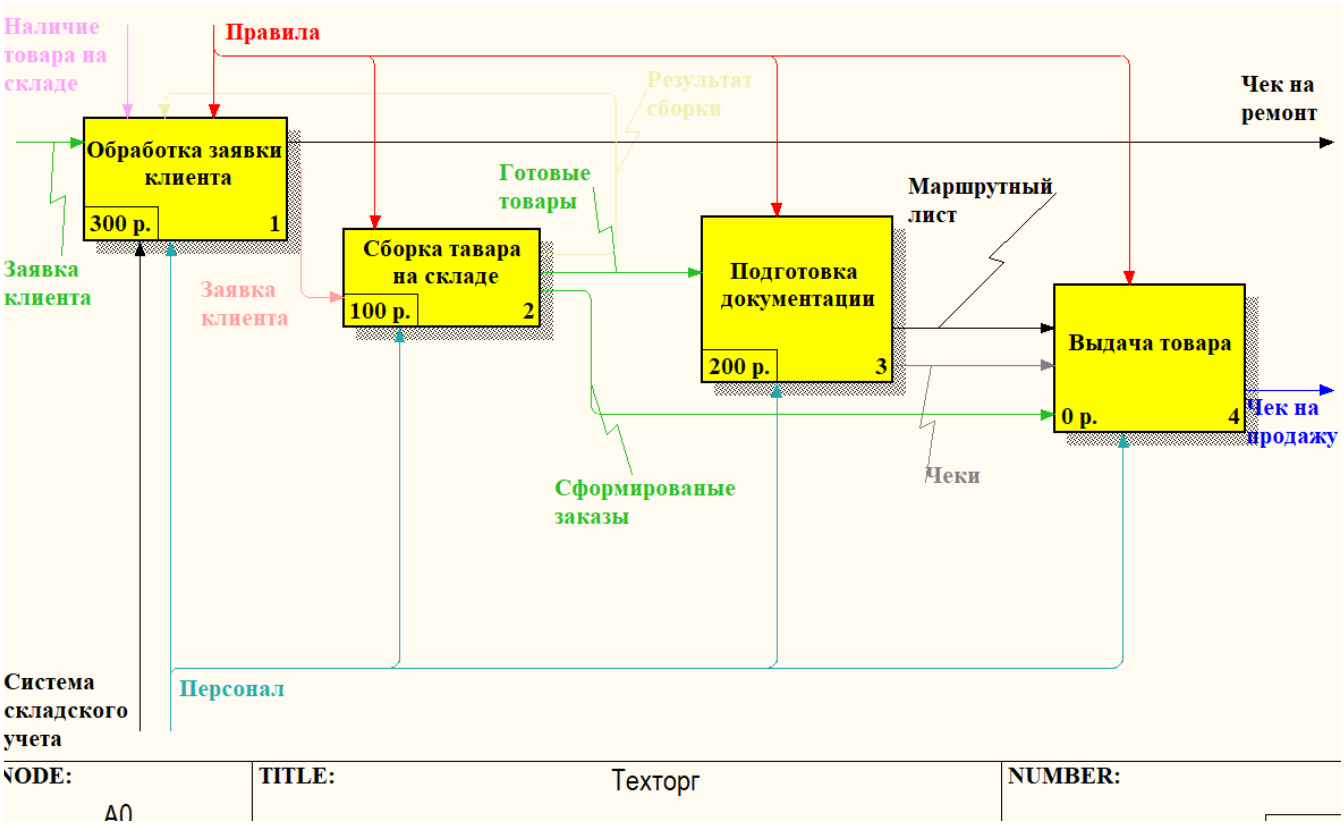


Рисунок 1.5 – Декомпозиция IDEF0 Техторг A0

На рисунке видно, что работа фирмы заключается в том, что клиент обращается к персоналу, после этого идет обработка заявки и сборка товара и подготовляется документация по товару, и производится выдача товара. На рис. 1.6 декомпозиция процесса обработки заявки.

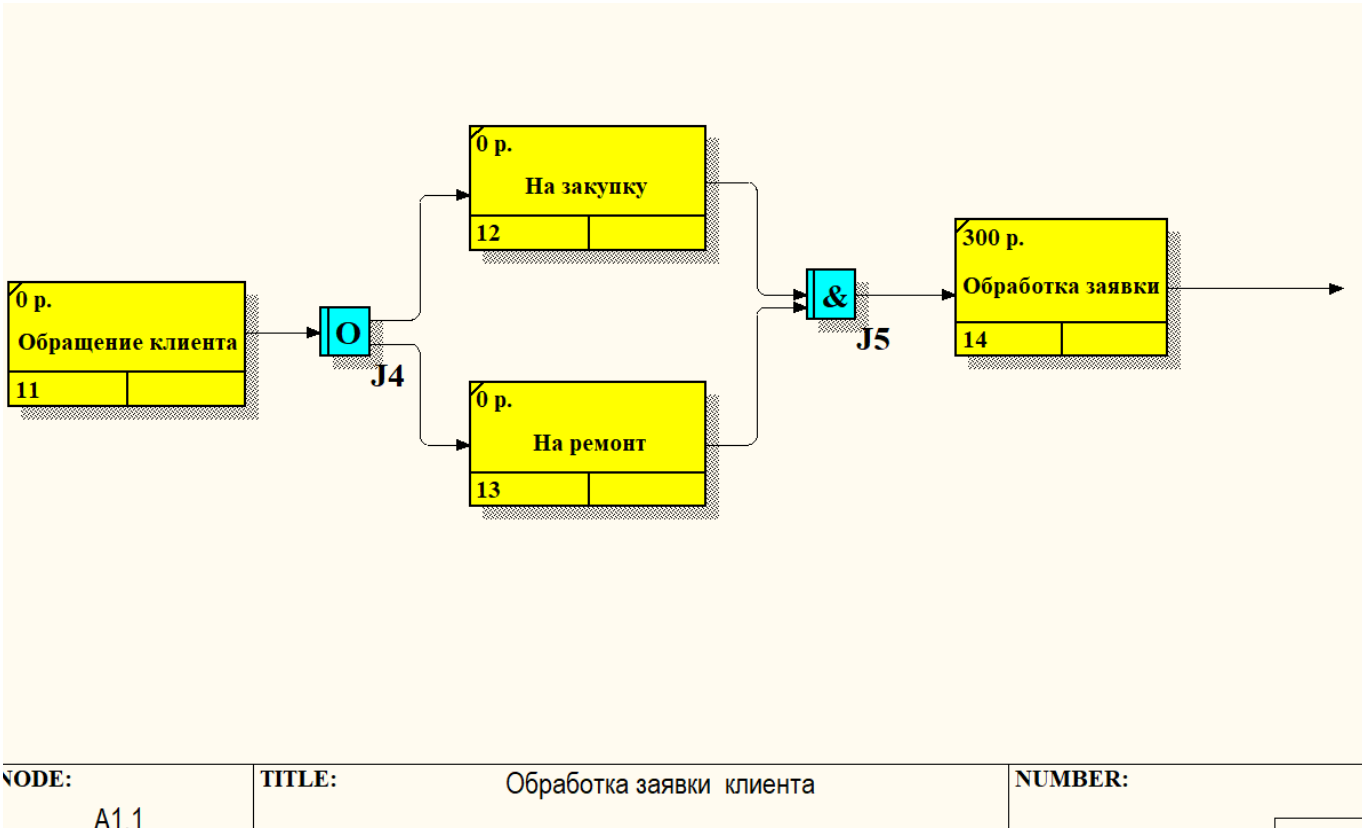


Рисунок 1.6 – Обработка заявки клиента A1.1

На рисунке более подробно видно, что клиент может подать заявку на ремонт или покупку товара. И после идет обработка заявки. На рис. 1.7 декомпозиция процесса сборки товара на складе.

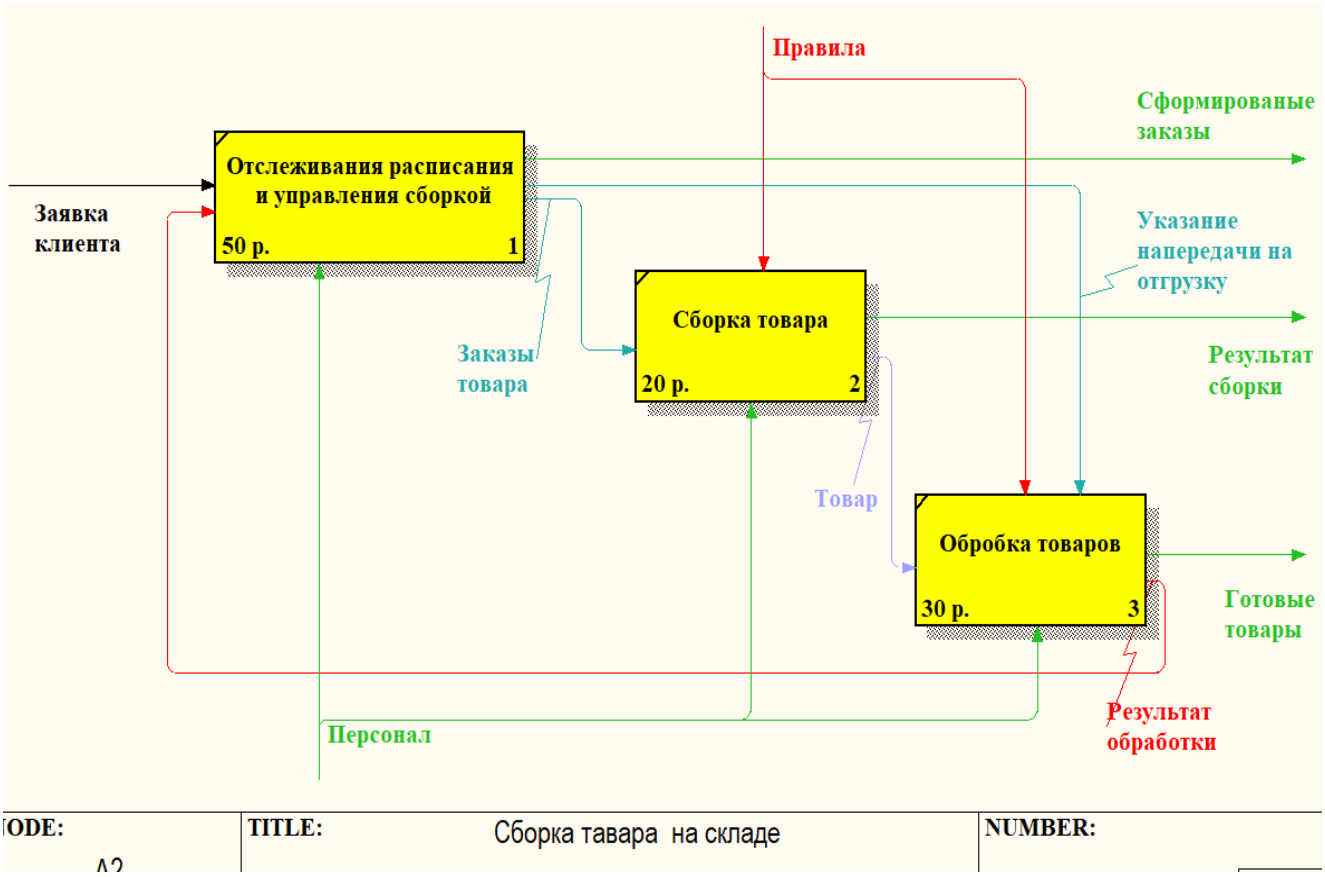


Рисунок 1.7 – Сборка товара на складе А2

На рисунке видно, что при сборке товара персонал отслеживает товар и после обрабатывает заказ. На рис. 1.8 показана выдача товара.

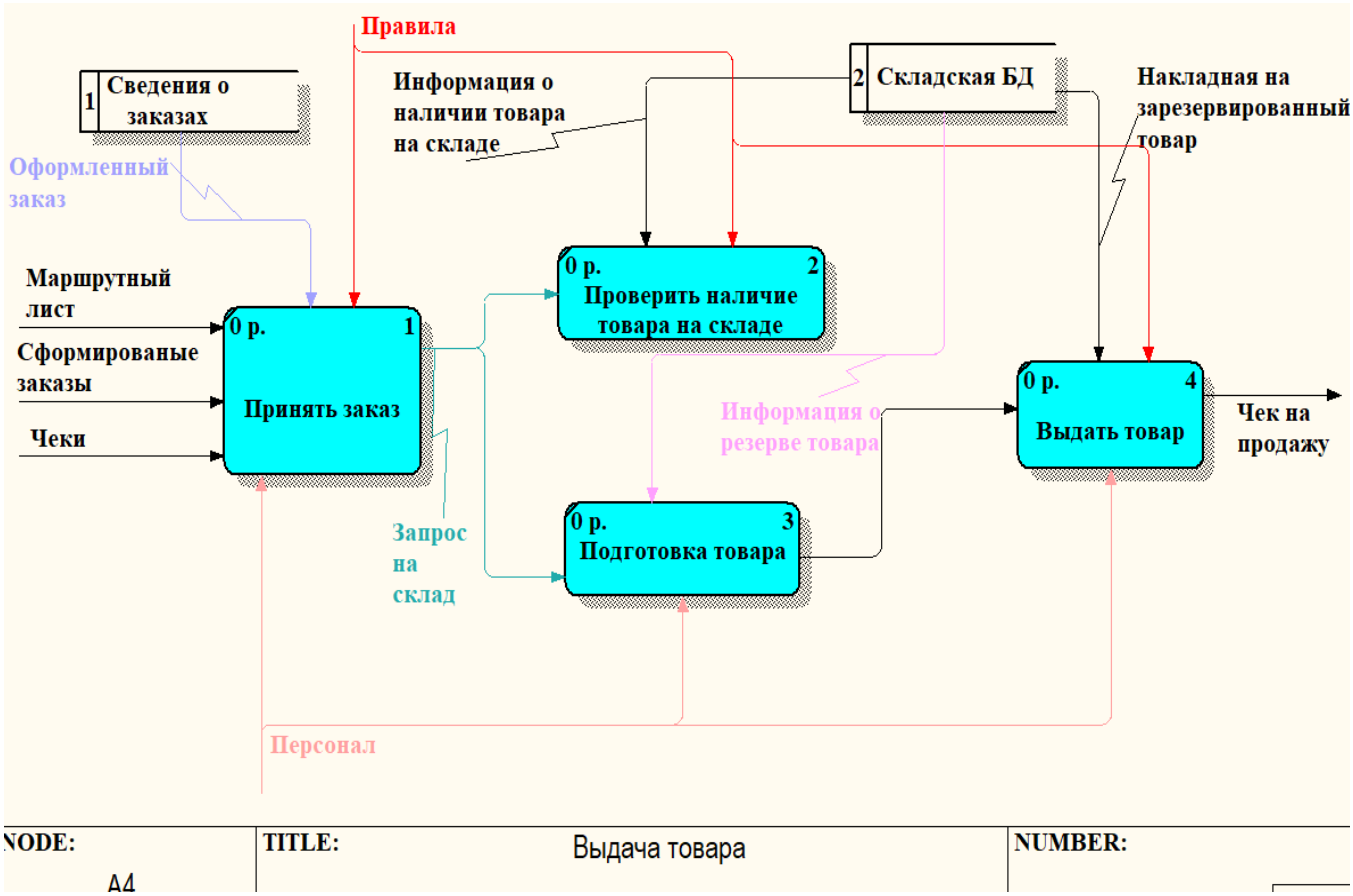


Рисунок 1.8 – Выдача товара А4

На рисунке показана выдача товара, при наличии чеков сотрудники принимают и подготавливают товар к выдаче. [1]

1.2.2. Определение места проектируемой задачи в комплексе задач и ее описание

Из всего комплекса задач по автоматизации работы «Техторг» мною будет исследоваться работа по обработке документов. Документы составляет менеджер по работе. По зарплате сотрудников, по учету заказов, по ремонту, по работе персонала, по товарам.

Входными данными для обработки всех документов являются:

- ФИО сотрудника;
- ФИО клиента;
- заказы;
- склад;

- товары;
- поставщики;
- поставка;
- филиал;
- ремонт.

Выходными информационными потоками в данном процессе являются:

- отчет по работе;
- отчет по заказам;
- отчет по поставке;
- доход;
- расход;
- чистый доход.

На рис. 1.9 показан процесс подготовки документации.

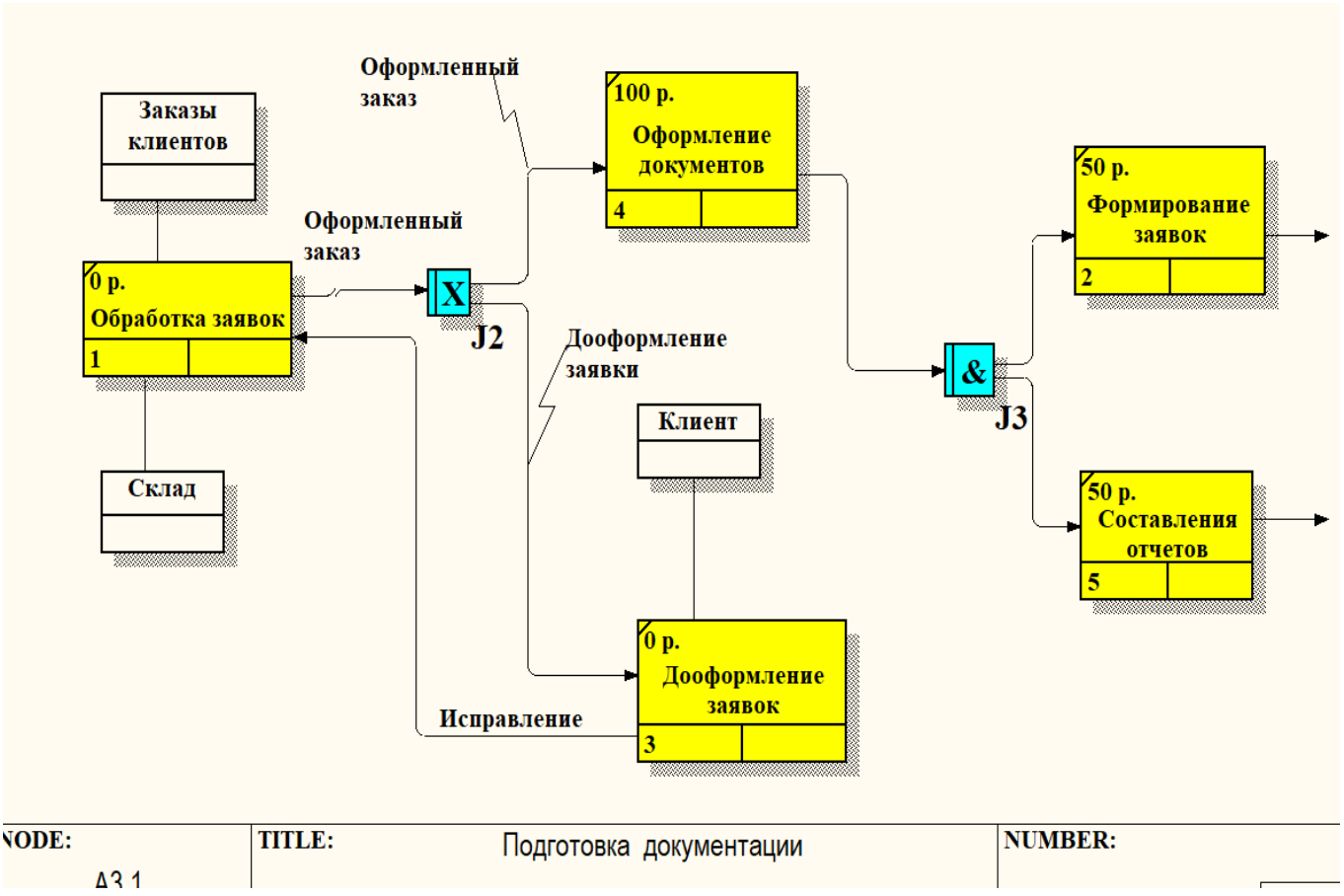


Рисунок 1.9 – Подготовка документации А3.1

На рисунке видно, что при получении заявки от клиента, идет оформление документов или изменение, при правильном оформлении идет формирование и составление отчетов.

Регламент работы задействованных специалистов:

На первом рубеже инженер выбирает задание реализации плана автоматизации и согласовывает её с системным админом.

На втором рубеже уже наличествует точное представление о том, чего желаем получить от доработки по автоматизации ИС. ТЗ передаётся разработчику программного обеспечения и сообщает программный код, а системный админ делает среду и сервисы для сопоставимости модуля и ИС. [2]

На третьем рубеже случается испытание перечня возможностей и лишь только впоследствии сего случается пилотное внедрение модуля. В ходе данной стадии выявляются промахи и вырабатываются свежие запросы или же советы.

Конечный четвёртый период предполагает возврат на 2 период, доработку перечня возможностей и поправка промахов в программном модуле и проведение финишного внедрения с извещением управления и всех юзеров системы. По мере надобности свежие функции ИС станут дорабатываться, применяя итерационный расклад. [2]

1.2.3 Обоснования необходимости использования вычислительной техники для решения задачи

Внедрение вычислительной техники при решениях комплекса задач, описываемого в предоставленной работе, обуславливается такими моментами. Размер и качество выходной информации не позволит решать задачи без применения вычислительной техники проворно и, собственно, что принципиально, тактично. Необходимость постоянной связи с различными юридическими базами данных, возможность использования локальной вычислительной сети, средств телекоммуникации и другие факторы, определяющие методы решения поставленных задач с использованием вычислительной техники.

В настоящее время роль компьютерной техники в деятельности предприятий продажи различной техники невозможно переоценить. На замену большим

книгам записи приходят резвые и необъёмные базы данных. Вместо выписки счета в несколько позиций вручную, документ оформляется компьютером в несколько секунд. Компьютер способен контролировать все денежные процессы и делать это намного лучше человека. Естественно, что для функционирования компьютера необходимо программное обеспечение. И если системное программное обеспечение на сегодняшний день не имеет особо широкого разнообразия для конечного пользователя, то на рынке прикладного программного обеспечения наблюдается довольно жесткая конкуренция. На фоне борьбы крупных программных корпораций за конечного пользователя единичные программные продукты просто незаметны. [3]

На сегодняшний день на предприятии уже существует комплекс вычислительной техники, предназначенный для обработки прихода и расхода товаров, оформления документов и прочей информации. Имеющаяся база данных дает информацию для принятия решений, но она не полная или не сгруппированная, что приводит в свою очередь к определенным трудностям в принятии стратегических решений.

Исходной первичной информацией в данном случае будут являться документы, предоставляющие количественно-стоимостную информацию (счета, расходные накладные, договора на поставку товара) и денежную информацию (платежное требование, счета, счета фактуры).

Сотрудник в своей работе использует программные продукты Microsoft Office. Для оформления договоров, приказов Microsoft Word, для работы с базой данных, складом Microsoft Excel. Использование этих программных средств не отвечает требованиям.

– Ручной ввод данных приводил к тому, что небольшая ошибка приводила к неверным расчетам, и ее приходится искать, просматривая огромное количество материала.

– Сложность поиска продукции. При заполнении расходных, приходных накладных и других документов, приходится искать в огромном списке товаров, которые есть на складе.

Все эти недостатки приводят к необоснованным затратам времени и ресурсов. Учет реализации товаров осуществляется по мере его отгрузки и

предъявления покупателям расчетных документов. Оформление документов по заказам и продаже товаров показаны на рисунке 1.10

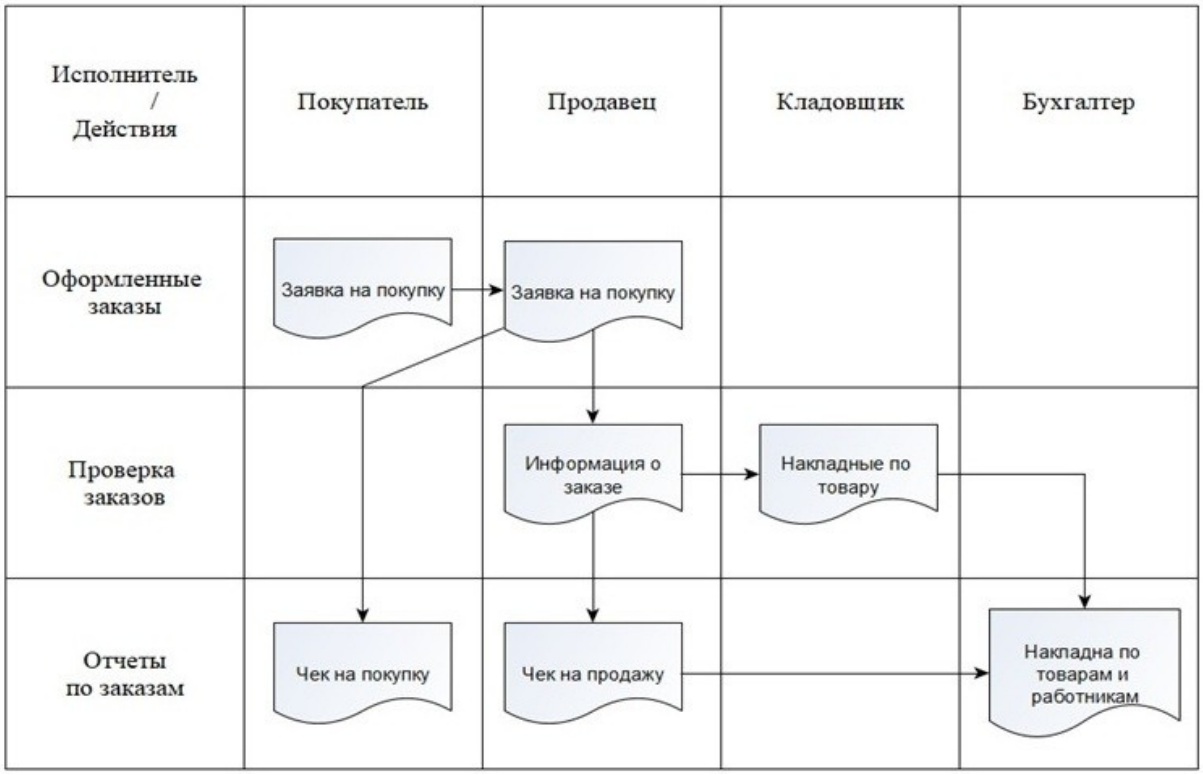


Рисунок 1.10 – Документооборот предприятия

Существующий способ получения заказов и метод учета связан с большой сложностью, фрагментарностью информации, что, скорее всего, приведет к их утрате или неправильному толкованию. На сегодняшний день невозможно получить информацию об общем количестве заказов, проанализировать основные причины возникновения проблемных вопросов у клиентов и проанализировать причины обращения.

Временные характеристики описанных процессов приведены в таблице 1.3

Таблица № 1.3

Действие	Среднее количество за рабочий день	Время, необходимое для выполнения одного действия, минут	Общее время, минут
Регистрация заявки	12	15	180
Поиск необходимой информации	6	5	30
Анализ информации за период	0.5	60	30
ИТОГО, минут:			240

Таким образом, каждый день, в среднем, 240 минут или 4 часа, сотрудник занят занесением различных сведений в книги учета, а также, при необходимости анализом и поиском нужных данных. Учитывая, что продолжительность рабочего дня составляет 8 часов, мы приходим к выводу, что на выполнение остальных обязанностей (то есть непосредственную работу по решению проблем клиента и выработке необходимых мероприятии) остается около 40 % рабочего времени, что крайне неэффективно.

В связи с этим, результатом проводимой автоматизации предполагается всегда иметь точнейшие сведения о количестве заявок, их видах, сократить время на создание аналитических отчетов и передачу документов в электронном виде. Очевидно, что для автоматизации требуется использовать такие средства, как персональные компьютеры, принтеры, а также специализированное программное обеспечение и, возможно, локальную вычислительную сеть.

Проведем расчетный тест ожидаемого эффекта от внедрения средств автоматизации. В данном случае при использовании вычислительной техники весь процесс значительно ускоряется, все необходимые данные будут внесены в общую БД и поиск нужной информации будет производиться при вводе необходимых параметров. В таблице 1.4 произведены расчеты эффекта внедрения.

Таблица № 1.4

Действие	Среднее количество за рабочий день	Время, необходимое для выполнения одного действия, минут	Общее время, минут
Регистрация заявки	12	3	36
Поиск необходимой информации	6	1	12
Анализ информации за период	0.5	5	2.5
ИТОГО, минут:			50.5

Таким образом, ожидаемая экономия рабочего времени составляет около 3 часов ежедневно, что позволяет увеличить эффективность работы сотрудников ООО «Техторг».

Также при использовании данного ПО будет иметься возможность составления отчетности на предприятие, выявление задолженности. К тому же информационные системы способствуют повышению производительности труда работников. Такое расширенное сотрудничество обеспечивает бесперебойное выполнение и реализацию различных задач и поставленных целей.

1.2.4 Анализ системы обеспечения информационной безопасности и защиты информации

У фирмы ООО “Техторг” есть своя политическая деятельность для защищенности. Все работники при поступлении получают должностное руководство, в котором внятно описаны его права и прямые обязанности с точки зрения информационной защиты информации. Любому работнику присваивается логин и пароль в системе доменной авторизации, в рамках которой выдается доступ к конкретным ресурсам приложения.

Защита информации на ПК в фирме осуществляется только программно. Осуществлено ограничение на включение новых периферийных приборов по

портам USB, com, ltp и PCI, она исполнена при помощи программного продукта ZLOCK и управляется централизовано на сервере. Для того чтобы подключить другой компонент к компьютеру, надо получить доступ администратора.

Хранение бумажных документов или файлов, а также флэш-накопителей и резервных копии, содержащие информацию, позволяющую установить личность, находятся в закрытой комнате или в закрытой картотеке. Также ограниченный доступ у сотрудников с законной деловой потребностью. Все контролируются, у кого есть ключ.

Вся конфиденциальная информация зашифровывается, даже та, которая отправляется третьим сторонам через общедоступные сети (например, Интернет).

Защита информации в серверной регламентируется следующими ограничениями доступа:

1) Права доступа по электронному пропуску (proxy- карта)

2) Роспись за получение и сдача ключа у охранника.

3)Разблокировка датчиков открытия двери у оперативного дежурного СБ

Средства защиты от инсайдерских угроз в компании, следующие:

За каждым пользователем закреплён компьютер/ноутбук и монитор, которые имеют идентификационный ПИН-код, за который он несет материальную ответственность.

1.3 Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ»

1.3.1 Анализ существующих разработок для автоматизации задачи

На сегодняшний день на рынке ПО есть множество разнообразных как иностранных, так и Русских услуг, и товаров по автоматизации документооборота фирм, как к примеру традиционный 1С: Документооборот.

В узком смысле информационной системой называют только подмножество компонентов ИС в широком смысле, включающие базы данных, СУБД и специализированные прикладные программы. ИС в узком смысле рассматривают как программно-аппаратную систему, предназначенную для автоматизации

целенаправленной деятельности конечных пользователей, обеспечивающую, в соответствии с заложенной в нее логикой обработки, возможность получения, модификации и хранения информации. [4]

Аппаратное обеспечение состоит из устройства ввода / вывода, процессора, операционной системы и мультимедийных устройств. Программное обеспечение состоит из различных программ и процедур. База данных состоит из данных, организованных в необходимой структуре. Сеть состоит из концентраторов, средств связи и сетевых устройств. Люди состоят из операторов устройств, сетевых администраторов и системных специалистов.

Обработка информации состоит из ввода; обработка данных, хранение данных, вывод и контроль. На этапе ввода, данные передаются в системы, которые на этапе обработки обрабатываются программным обеспечением и другими запросами. На этапе вывода данные представляются в структурированном формате и в отчетах. [5]

Перед выбором любой информационной системы всегда встаёт вопрос о требованиях к системе и целях, которые планируется достигнуть в результате внедрения. На текущий момент в исследуемой мной компании ООО “Техторг” уже внедрена ИС 1С: Документооборот и используется более 2 лет, за это время были осознаны некоторые ошибки внедрения и появились некоторые требования по доработке системы. Итак, основными критериями при выборе технологии решения задачи будут следующие условия:

- суммарный бюджет на проект автоматизации не более 100 000 руб.;
- сокращение времени по добавлению новых сотрудников;
- добавление новых функций, такие как, просмотр доходов филиала и расходов, также статистику продаж товаров и др.;
- контроль над обработанными заявками в автоматическом режиме.

1С: Документооборот

Система электронного документооборота может работать в единой базе с продуктом "1С: Зарплата и управление персоналом" или "1С: Управление производственным предприятием", что позволяет комплексно решать и функциональные задачи (например, набор персонала), и учетные (начисление оклада) и управленческие (принятие мотивационного решения о повышении оклада или

штрафе), и задачи взаимодействия и передачи информации (прохождения документа по маршруту, связанное со своевременным оповещением всех заинтересованных участников процесса согласно ранее утвержденному порядку).

Megapolis.DocNet

Все документы предприятия онлайн с новыми возможностями:

- Одновременная работа более 60 тыс. пользователей;
- Электронный архив до 100 Тб;
- Работа через интернет браузер или мобильное приложение;
- Защита данных повышенного уровня секретности;
- Открытое API для интеграции со сторонними информационными системами.

Основные модули выполняют задачи регистрации всех типов документов, сканирование и распознавание текстов, параллельного и последовательного согласования, юридически значимого подписания документов ЭЦП или КЭП, создание шаблонов, уведомления, архивирования и контроля полного жизненного цикла документа и работы с ним.

Foss Doc

Система электронного документооборота Foss Doc – решение на платформе Foss Look, предназначенное для создания электронного архива документов, организации корпоративного документооборота (workflow) и автоматизации бизнес-процессов на предприятиях, в учреждениях и организациях любого рода деятельности. Программа позволяет решить большое количество задач, реализация которых возложена на соответствующие модули. Система может быть легко перенастроена с учетом специфики работы каждого конкретного предприятия.

Основные характеристики предоставленных продуктов представлены в таблице 1.5

Таблица № 1.5

Характеристики информационных систем

Программа / Характеристики	**1С: Документооборот**	**Megapolis.Doc Net**	**Foss Doc**
Стоимость	Клиентская лицензия на 100 рабочих мест 459 000	0т 220 000	от 120 000
Возможность доработки своими силами	Имеется возможность настройки взаимодействия с другими приложениями и обмена данными через текстовые файлы, XML-документы и DBF-файлы, и WEB-сервисы	Решение состоит из встроенных моделей и компонентов, которые легко настраиваются, позволяют быстро конфигурировать сложные системы.	Есть возможность кастомизировать (настроить под себя) ваш продукт с помощью функциональности библиотек документов: добавить или удалить поля и/или функции документов, настроить справочники.
Интерфейс	Интуитивно понятный, лишний функционал отключается.	Перегружен, спроектирован с ориентированием на документооборот	Интуитивно понятен, отсутствует дополнительный функционал, плохая масштабируемость.
Кроссплатформенность	Присутствует(Windows, Linux, Mac)	Отсутствует, только Linux	Присутствует, только Windows

Исходя из критериев, которые предъявляет компания ООО “Техторг” было решено создать новую, дешевле и более функциональную систему автоматизации по документообороту.

1.3.2 Выбор и обоснование стратегии автоматизации задачи

Определим стратегические и функциональные свойства ИС. Внедряемая ИС автоматизации должна обладать хорошей масштабируемостью и высокой надежностью хранения данных. Так же система должна легко адаптироваться под программную среду объекта внедрения. При внедрении, данная ИС должна иметь возможность гибкой доработки, так как возможно потребуется дальнейшая доработка ИС силами сотрудников ИТ. Система не должна быть загромождена большим количеством ненужных функций. Понятный интерфейс пользователя, функциональный интерфейс администратора и широкую систему отчетов. [6]

Для реализации плана возможно отметить надлежащие рубежи:

1. Анализ имеющих место быть процессов учёта работы ИТ отдела фирмы. Заключается в сборе и анализе надлежащих характеристик: количество не произведенных заказов, численность произведенных заказов в день по инженеру, среднее время выполнения заказа.
2. Определение абсолютной перечни возможностей, которых надо будет автоматизировать. Определение того что уже есть в системе в данный момент и тех функций, которые станут доступны в процессе разработки и внедрения ИС.
3. Выбор стратегии автоматизации. Тест имеющихся типов стратегий автоматизации и подбор более благоприятной к структуре и бизнес процессам фирмы.
4. Формирование технического поручения и графика выполнения заданий.

Выбор стратегии автоматизации:

Существует четыре варианта стратегии автоматизации: кусочная (хаотичная) автоматизация, автоматизация по участкам, автоматизация по направлениям и комплексная автоматизация. Кусочная автоматизация предполагает под собой приобретение предприятием без конкретного стратегического плана отдельных фрагментов информационной системы, которые не способны оказать реальной

пользы предприятию в целом. Дальнейшее развитие информационной системы предприятия связано с новыми, значительными затратами.

Автоматизация по участкам учитывает автоматизацию отдельных производственных участков, соединенных по набору выполняемых функций. Данный метод автоматизации выбирается при условии, в случае если есть участки, где использование автоматических систем выделяет важный финансовый эффект, к примеру, за счет уменьшения персонала.

Автоматизация по направлениям подразумевает под собой автоматизацию отдельных направлений деятельности компании. В этом случае компания получает полную автоматизацию работы, например, кадровой службы, производства, бухгалтерии или логистики. Такой подход к автоматизации вполне нормален и в дальнейшем интеграция уже автоматизированных направлений в рамках всего предприятия не будет связана с серьезными препятствиями. [7]

Хаотичная, она же кусочная стратегия не подходит, например, как мы автоматизируем весь отдел, в данном случае применяется некоторое количество элементов ИС, такие как интернет сервер, СУБД, серверное приложение, невозможно автоматизировать лишь только долю из них, эффекта от сего станет ноль.

Полная стратегия автоматизации тоже не подходит в нашем случае, так как мы автоматизируем одно из направлений деятельности компании.

Стратегия автоматизации по участкам лучше других подходит в данном дипломном плане, вследствие того собственно, что в нашем случае мы автоматизируем отдел логистики, за смену в нем трудится 5 человек, которые 30 % времени заняты телефонными звонками, а всё остальное время заняты рутинной работой по регистрации и обработке заказов. Автоматизировав этот процесс, возможно, станет уменьшено количество нагрузки на предоставленный отдел, не понижая продуктивности его работы.

1.3.3 Выбор и обоснование способа приобретения ИС для автоматизации комплекса задач

Для автоматизации процессов управления и информационного обеспечения рассматриваются три способа приобретения программного обеспечения:

- Покупка готовой специализированной ИС;
- Разработка ИС своими силами;
- Разработка ИС сторонней фирмой;
- Покупка системы и её доработка;

Ниже приведена таблица 1.6, в которой сравниваются критерии отбора, способа приобретения программного продукта.

Таблица № 1.6

Сравнение способов приобретения ИС

Способ приобретения \ критерии	**Покупка готовой специализированной ИС**	**Разработка ИС своими силами**	**Разработка ИС сторонней фирмой**	**Покупка системы и её доработка**
Соответствие поставленной задаче	Несоответствие поставленной задаче, за счет невозможности автоматизации собственных бизнес процессов	Полное соответствие требованиям к системе	Частичное соответствие потребностям	Полное соответствие требованиям к системе
Стоимость внедрения	До 100000 рублей	До 200000 рублей	От 300000 рублей	До 150000 рублей
Адаптивность	Невозможность изменения системы	Возможность полной переработки	Возможность изменения силами компании-разработчика	Возможность переработки разрабатываемых процессов

Продолжение таблицы № 1.6

Способ приобретения \ критерии	**Покупка готовой специализированной ИС**	**Разработка ИС своими силами**	**Разработка ИС сторонней фирмой**	**Покупка системы и её доработка**
Надежность	Надежность гарантируется фирмой производителем ИС	Слабая надежность ИС	Надежность гарантируется фирмой-разработчиком ИС	Высокая надежность ИС

Покупка: Так называемую коробочную версию продукта в дипломном проекте рассматривать не буду, так как там нечего проектировать.

Покупка плюс доработка: Это покупка уже готового решения с возможностью дописать функционал силами самих производителей ПО либо сделать это средствами своей компании, то есть программистом, работающим в компании на постоянной основе, в этом случае язык программирования программиста должен совпадать или хотя бы быть близок к тому языку программирования, на котором пишет программист компании.

Разработка: В данном случае можно использовать внешних разработчиков, предоставив им чёткое техническое задание или разрабатывать ПО силами ИТ специалистов, работающих в компании. В обоих случаях это весьма трудозатратный проект с внедрением, который займет не менее чем полгода.

Приобретение плюс доработка были избраны исходя из критериев, описанных в пункте 1.3.1 т.к. на этот момент применяемая ИС внедрена не наиболее наилучшим образом и ведущей процесс регистрации и обработки заказов мало автоматизирован. Плюсом данного заключения станет вероятность параллельной работы в системе без кардинальных перемен во время доработки. Этот вариант нормально подходит еще вследствие того, собственно, что в случае если что-нибудь происходит не так, то всякий раз возможно остановится, изучить и начать интеграцию снова, совместно с тем, главная система станет продолжать функционировать исправно. Еще не считая способности откатиться в начало, в

критериях не всеобъемлемости бюджета, этот вариант станет единственным правильным заключением.

1.4 Обоснование проектных решений

1.4.1 Обоснование проектных решений по информационному обеспечению

В данном дипломном проекте будет использоваться только локальный классификатор, так как никаких других классификаторов РФ в системе не используется. Классифицировать будем заявки по приоритетности заявки, по компании и по зоне ИТ, к которой принадлежит эта заявка (тип заявки).

Значительную долю вне машинного ИО составляет документация. В условиях автоматизации важное значение придается унификации документации, устанавливающей единые требования к содержанию и построению документов. Унифицированные формы документов вырабатываются как для всех предприятий РФ (например, формы бухгалтерской отчетности), так и для отдельных предприятий (например, формы управленческой отчетности). Унификация заключается в тщательном отборе и четком определении необходимой номенклатуры документов. При этом определяются сферы назначения и использования документов и выявляются специфические особенности, характерные для соответствующих видов документов. Документы могут быть унифицированными и локальными.

В данном дипломном проекте использованы локальные документы: «Заявка на закупку материальных ценностей», «Заявка на регистрацию пользователя и доступ к программным ресурсам», «Заявка на устранение неполадок», «Заявка на доступ к сетевым ресурсам». Информационные файлы формируются на основе исходной информации, содержащейся в вышеуказанных первичных документах - основных носителях первичной экономической информации в системах машинной обработки данных. К ним предъявляется ряд требований:

- достаточная полнота информации для решения задачи;
- исключение избыточности информации;
- достоверность и своевременность информации;

- согласованность форм первичных документов с макетами размещения информации на машинном носителе;
- логичность построения документа.

Существует три способа организации информационной базы (ИБ): файловая организация ИБ; интегрированная ИБ и смешанная организация ИБ.

Под файловой организацией ИБ понимается локальное размещение базы на компьютере, доступ к которому других пользователей осуществляется стандартными методами ОС для обмена данными по сети, например, в MS Windows это Sharing и Security, что уменьшает скорость обработки данных в локальной базе. Под смешанной организацией ИБ подразумевается распределённая база данных, хранящаяся на нескольких серверах и реплицирующая изменения в каждой из них по расписанию, данная структура ИБ используется в системах класса ERP для работы в одной ИБ территориально удалённым офисам одновременно.

В данном дипломном проекте наиболее целесообразной организацией ИБ считаю интегрированную организацию ИБ, так как размер базы будет увеличиваться каждый день на 100-200 записей. И оптимальным выбором будет использование СУБД вместо файлового хранения базы данных.

1.4.2 Обоснование проектных решений по программному обеспечению

Программное обеспечение – неотъемлемая часть компьютерной системы. Оно является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него программным обеспечением. Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах. [8] В отличие от аппаратного обеспечения, которое описывает физические аспекты компьютера, программное обеспечение - это общий термин, используемый для обозначения приложений, сценариев и программ, работающих на устройстве. Программное обеспечение можно рассматривать как переменную часть компьютера, а аппаратную часть как неизменную часть.

Программное обеспечение часто подразделяется на прикладное программное обеспечение или загружаемые пользователем программы, которые

удовлетворяют необходимые нужды или потребности, и системное программное обеспечение, которое включает в себя операционные системы и любые программы, поддерживающие прикладное программное обеспечение. Термин промежуточное программное обеспечение иногда используется для описания программирования, которое является посредником между прикладным и системным программным обеспечением или между двумя различными типами прикладного программного обеспечения. Например, промежуточное программное обеспечение может использоваться для отправки запроса удаленной работы из приложения на компьютере, на котором установлена операционная система одного типа, на приложение на компьютере с другой операционной системой.

Дополнительной категорией программного обеспечения является утилита, представляющая собой небольшую полезную программу с ограниченными возможностями. Некоторые утилиты поставляются с операционными системами. Как и приложения, утилиты, как правило, устанавливаются отдельно и могут использоваться независимо от остальной части операционной системы. Критериями выбора ПО, установленного на ПК пользователей является максимальная минимизация времени процесса описание заявки/неисправности пользователей и отправки заявки используя корпоративную почту компании. В рамках корпоративного стандарта в компании используется MS Outlook 2010. Для запуска этой программы на компьютере пользователя должна быть также установлена ОС, поддерживающая запуск данного приложения. В связи с корпоративным стандартом используется версия ОС MS Windows 10 PRO, так как на данный момент, это единственная ОС поддерживаемая компанией Microsoft в России (остальные версии на данный момент уже не поддерживаются) и имеющая возможность работать в доменной инфраструктуре (версия Windows 10 Home не поддерживает работу в домене). Исходя из этих обоснований критерий Win 10 Pro Rus будет использована в качестве пользовательской ОС. Итог: на рабочих станциях пользователей (клиентов) должны быть установлены следующие программные продукты:

- Операционная система Windows 10 Pro Rus
- Почтовый клиент MS Outlook 2010

Программа по обработке документооборота – это программа, разработанная в ходе написания дипломного проекта на языке программирования C#, которая реализует весь документооборот на фирме. Критериями выбора архитектуры реализации проектируемого Программного Комплекса являются:

- совместимость с существующей инфраструктурой серверов;
- возможность создания резервных копий данных и просматривать статистку по обработанным заявкам.

При выборе ОС под данное решение был выбран MS Windows Server 2017, так как данный вид серверных ОС не сильно требователен к ресурсам, а также давно используются в инфраструктуре компании.

Для серверной части необходимо также выбрать систему управления базами данных (СУБД). В таблице 1.7 представлены основные современные СУБД.

Таблица № 1.7

Варианты СУБД систем.

Название СУБД **Характеристики**	**Microsoft SQL Server 2017**	**MySql**	**Oracle 12.1.0.1**
Встраиваемая аутентификация пользователей	да	да	нет
Мониторинг работы базы данных	да	да	да
Возможность создание временные таблицы	да	да	да
Ведение журнала действий с базами данных и подключение	да	да	да

В качестве СУБД для программы будет использоваться Microsoft SQL Server 2017. Выбор в пользу компании Microsoft сделан не случайно, потому что язык разработки выбран тоже компании Microsoft.

Закупка нового сервера не обязательна, так как база данных программного продукта может быть расположена на уже имеющемся сервере. При создании данного сервера компания закладывала около 50 процентов мощности

закупаемого сервера на будущую масштабируемость системы, но не смотря на рост базы данных загруженность сервера возросла всего на 15 процентов.

В рамках уже существующей базы данных в ней будут созданы дополнительные таблицы и связи между ними. Добавленные таблицы не будут на прямую связаны с основными функциональными таблицами базы данных.

Рассмотрим основные методы и средства проектирования. В таблице 1.8 сравниваются между собой основные средства проектирования.

Таблица № 1.8

Основные средства проектирования.

Название **Характеристики**	**Microsoft Visual C# (MSVC)**	**Delphi 2010**	**Microsoft Visual C++ (MSVC)**
Совместимость с операционной системой Windows	да	да	да
Встроенный редактор интерфейса	да	да	да
Совместимость с другими языками программирования	да	нет	да

Написание программы сервиса, обрабатывающего документооборот предприятия, было решено на языке C#. Такой выбор был сделан не случайно. Во-первых, администратор предприятия имеет опыт работы с данным языком программирования. Во-вторых, данный язык действительно не сложен в освоении и среди фрилансеров в интернете легко найти программиста, который при правильно поставленном техническом задании сможет в поставленные сроки выполнить данную разработку, стоимость разработки ПО на C# гораздо ниже, чем разработка на основе таких громоздких продуктов как C++ или Delphi. В-третьих, данный язык очень активно развивается и имеет открытую архитектуру и легко позволяет сделать графический интерфейс в виде виндовс форм.

1.4.3 Обоснование проектных решений по техническому обеспечению

Техническое обеспечение - это совокупность технических средств, компьютерной техники, средств передачи информации, используемых в автоматизированных системах управления и в информационных системах.

Для функционирования программы обработки документации на предприятии потребуются следующие элементы технического обеспечения:

- ПК-сервер – это основная ЭВМ, на которой будет создана сама база данных в СУБД, а также данный сервер будет являться сервером БД. На нем же будет размещена база данных. Характеристики данного сервера представлены в таблице выше;
- ПК-пользователя (рабочая станция) – это пользовательский ПК, посредством которого будут создаваться запрос на получения указанного запроса;
- средства организации ЛВС – в данный перечень входят активные (маршрутизатор, коммутатор, шлюз и тд.) и пассивные (сегменты ЛВС, коммутационные розетки и тд.) компоненты локальной вычислительной сети.

Для каждого элемента мы выберем несколько критериев, наиболее критичных при осуществлении выбора.

Серверы - это самые незаметные системы в целой сети компьютеров. Их прячут от посторонних глаз, не выставляя напоказ, но берегут, как зеницу ока. Идеальный сервер - система, которая стоит в уголочке или в шкафу-стойке, и о ней все забыли. Этим он отличается от обычных ПК или рабочей станции. И подход к выбору сервера гораздо более жесткий и прагматичный, чем к любой другой системе. При этом специфика сервера - преднамеренная избыточность основных компонентов. И горе тому, кто пренебрежет этой "природной сутью" его в попытках сэкономить.

Главными критериями выбора серверной платформы являются специфика решаемых сервером задач и количество автоматизированных рабочих мест, которые объединяются в сеть. После этого остается только выбрать производителя.

Для Сервера СУБД в рамках одного ПК основным критерием выбора будет отказоустойчивость и пропускная способность сетевого интерфейса. Исходя из среднего арифметического числа заказов в день, которое оформляет 500

заказов с ежеминутным выборочным опросом программы базы данных на присутствие свежих заказов в основе, а еще загруженности сетевой инфраструктуры на 30 процентов и загруженности коллективного сервера баз данных на 25 процентов, нет надобности закупать высокопроизводительный сервер с сетевым адаптером скоростью в 1Gbps, довольно ограничиться интерфейсом в 100 Mbps. Итогом анализа критериев по серверному оборудованию будет использование того же сервера, который использовался в компании ранее. В качестве сервера баз данных (БД) используется сервер, построенный на платформе HP ProLiant DL365 G5, он обладает следующими характеристиками, представленными в таблице 1.9

Таблица № 1.9

Характеристики сервера БД.

Характеристика	**Значение**
Процессор	Двуядерный Intel® Xeon® X5260 с тактовой частотой 3,3 Гц.
Кол-во процессоров	8
Оперативная память	16 Гб (расширяемая до 64Гб)
Жесткие диски	Тип «SAS» 4 диска 300 Гб и 2 диска 74 Гб
Кол-во жестких дисков	6 (расширяемо до 8)
Питание	Дополнительно резервный блок питания 800 Вт с горячей заменой

Рассмотрим данную платформу подробнее:

- Два процессора позволяют, при использовании SQL сервера, эффективно распараллеливать задачи, выполняемые на сервере.

- 16 Гб оперативной памяти достаточно для обработки больших объемов информации, используемых на данный момент в БД, а также последующего увеличения вычислительной нагрузки, так как на данный момент пиковый размер занятой оперативной памяти составляет 6 Гб.

- Использование 6 жестких дисков обусловлено следующими соображениями:

Для достоверного функционирования операционной системы сервера санкционирован RAID массив из 2-ух строгих дисков объемом по 74 ГБ (такого размера довольно для работы ОС). Операционная система нарочно размещена порознь от файлов БД из мнения защищенности и производительности.

Для достоверного сбережения данных в формате Structured Query Language (SQL) санкционирован массив строгих дисков наибольшего размера 300 Гб. Предоставленного размера довольно для внедрения свежего перечня возможностей, на данный момент размер занятого места занимает 53Гб, при условии того собственно, что в БД хранятся записи за 3 года.

Так же отдельно необходимо хранить данные в форматах mdf (файл базы данных), а также транзакции в виде файлов ldf (файл транзакций), для чего необходим еще один массив, аналогичный предыдущему по размеру.

Использование дополнительного питания повышает отказоустойчивость системы.

Для ПК-пользователя и ПК-инженера основными критерием выбора оборудования в данном случае являются его технические характеристика, и что не менее важно, соответствие утвержденным корпоративным стандартам. Таким образом, для персональных компьютеров основным критерием выбора является утвержденный список конфигураций ПК. (B6000-ITX: Core i3-9100/ 8 Гб/ 1 Тб/ UHD Graphics 630) Характеристики данной конфигурации описаны в таблице 1.10

Таблица № 1.10

Характеристика конфигурации ПК пользователя.

Наименование	**Конфигурация**
Корпус	DeskTop INWIN BM639 <Black> Mini-iTX/Mini-DTX 160W (24+4пин)
Процессор	CPU Intel Core i3-9100 3.6 GHz/4core/SVGA UHD Graphics 630/1+6Mb/65W/8 GT/s LGA1151
Куллер для процессора	DEEPCOOL <DP-ICAP-T9P> THETA 9 PWM (4пин, 1155, 17.8-41.3дБ, 1100-3200об/мин, Al)
Вентилятор	GlacialTech "SilentBlade II GT9225-EDLA1" d90мм, 1600об./мин. (OEM)

Продолжение таблицы № 1.10

Наименование	**Конфигурация**
Материнская плата	GIGABYTE H310M S2V 2.0 (RTL) LGA1151 <H310> PCI-E Dsub+DVI GbLAN SATA MicroATX 2DDR4
Модуль оператив-ной памяти	Crucial <CT8G4DFS8266> DDR4 DIMM 8Gb <PC4-21300> CL19
Жесткий диск	HDD 1 Tb SATA 6Gb/s Seagate Barracuda <ST1000DM010> 3.5" 7200rpm 64Mb
Привод DVD±RW	DVD RAM&DVD±R/RW&CDRW ASUS DRW-24D5MT <Black> SATA (OEM)

II Проектная часть

2.1 Разработка проекта автоматизации

2.1.1 Этапы Жизненного Цикла проекта автоматизации

Жизненный цикл разработки программного обеспечения (ЖЦ) - это процесс, используемый индустрией программного обеспечения для проектирования, разработки и тестирования высококачественного программного обеспечения. ЖЦ нацелен на производство высококачественного программного обеспечения, которое соответствует ожиданиям клиентов или превосходит их, в кратчайшие сроки завершает работу и оценивает затраты.

- ЖЦ является аббревиатурой жизненного цикла разработки программного обеспечения.
- Это также называется процессом разработки программного обеспечения.
- ЖЦ - это структура, определяющая задачи, выполняемые на каждом этапе процесса разработки программного обеспечения.
- ГОСТ Р ИСО/МЭК 12207-2010 является международным стандартом для процессов жизненного цикла программного обеспечения. Он призван стать стандартом, определяющим все задачи, необходимые для разработки и обслуживания программного обеспечения.

ЖЦ это процесс, которому следует программный проект в рамках организации программного обеспечения. Он состоит из подробного плана, описывающего, как разрабатывать, поддерживать, заменять и изменять или улучшать конкретное программное обеспечение. Жизненный цикл определяет методологию улучшения качества программного обеспечения и общего процесса разработки. [9]

Custom Development Method (методика Oracle) по разработке прикладных информационных систем под заказ - конкретный материал, детализированный до уровня заготовок проектных документов, рассчитанных на использование в проектах с применением Oracle. Степень адаптивности CDM ограничивается тремя моделями ЖЦ: "классическая" (предусмотрены все работы/задачи и эта-

рекомендуемый в случае малых проектов и возможности быстро прототипировать приложения.

RUP был разработан на основе результатов сотни различных проектов по разработке программного обеспечения и идеи некоторых из наиболее влияющих на людей в программной инженерии дисциплины. XP представляет гибкие модели процесса. MSF была разработана на основе большого опыта Microsoft собранном на протяжении десятилетий.

RUP, процесс доступен как продукт, становится объектом частой практики в отрасли за последние годы и был расширен для поддержки элементов XP.

Microsoft продвигает свою Framework систему, включив ее в предстоящем выпуске системы Visual Studio Team и следует примеру Rational, предоставляя не только описание процесса, но и полный продукт для его поддержки.

Rational Unified Process является наиболее часто используемым процессом разработки программного обеспечения в промышленности. Основными преимуществами являются поддержка через Rational Software, которая постоянно совершенствует процесс, поддержка инструмента и плотно соединенную инструмент документацию, а также поддержку Rational для наставничества в реализации процесса. RUP обеспечивает хорошо структурированную основу, разделенную на фазы и рабочие процессы, что обеспечивает легкую навигацию в рамках структуры. Обеспечивает команде разработчиков свободный доступ к базе знаний с инструкциями для использования программных средств. Процесс включает в себя комплексные меры по обеспечению качества. Минимальные стандарты к требованиям проектирования и разработке программного итерационного обеспечения включены, а также тестирование, управление конфигурациями и взаимодействие с клиентом в течение всего процесса.

Framework Microsoft Solution обеспечивает глубокое понимание разработки программного обеспечения практикуемого в Microsoft. MSF представляет собой попытку распространять компанией Microsoft свои знания в области разработки программного обеспечения. MSF является одной из двух дополняющих друг друга структур (кроме Microsoft Operations Framework (MOF)), образуя комплексное решение для крупных софтверных компаний, занятых в средних и крупных проектах. Тем не менее, MSF может применяться самостоятельно. Она

не зависит от финансового отдела, что обеспечивает легкую реализацию в средних компаниях. Благодаря своей гибкости, масштабируемости и отсутствию жестких инструкций MSF способен удовлетворить нужды организации или проектной группы любого размера. Оба принципа не только улучшают качество программного обеспечения, но и то, как оно построено. Microsoft предоставляет несколько моделей шаблонов, рекомендованных в MSF. Поскольку MSF не включает в себя огромный набор рекомендуемых средств (которые будут меняться с следующей версией Visual Studio), она может быть легко интегрирована в существующую среду разработки. [10]

Экстремальное программирование представляет собой совершенно новый подход к разработке программного обеспечения. Для небольших проектов, XP предлагает хороший подход для достижения высокого качества программного обеспечения. Тесное вовлечение клиента, основное внимание на тестировании, и подход к снижению ИКР поддержки этой цели. Тем не менее, XP может привести к организационным проблемам при применении в больших проектах. Для большинства проектов, трудно достичь тесную интеграцию клиента в процессе разработки.

Основные критерии для выбора стандарта ЖЦ будут в таблице 2.1.

Итерация – для того, чтобы установить более высокое качество программного обеспечения, процесс разработки ПО должен использовать итеративный и инкрементный подход к развитию. Итерационные циклы включают все виды деятельности в области развития анализа, проектирования, реализации, тестирования и, наконец, развертывания. Контроль качества может применяться более эффективно в течение всего процесса. При использовании итеративного подхода, процесс приобретает большую гибкость при работе с изменяющимися требованиями или областью. Выпуски продукта ранней обратной связи от клиентов и заинтересованных сторон, которая имеет жизненно важное значение для улучшения общего качества программного обеспечения. Однако, итеративная разработка должна быть поддержана управлением рисками и ранним вовлечением конечных пользователей для достижения своего полного потенциала. XP основывается на очень строгом итеративном подходе, требующем ежедневную сборку всех компонентов. Это ограничивает время, необходимое для обнаружения

ошибок разработчиками и своевременного решения этих проблем. Конечно, неполные компоненты или отдельные методы исключены из ежедневной сборки. Структура декомпозиции работ должна рассмотреть эти вопросы, чтобы позволить интеграцию мелких компонентов каждый день. Использование этого подхода требует тщательного планирования, но, безусловно, обеспечивает высокое качество программного обеспечения.

Качество как цель – в процессе разработки программного обеспечения необходимо определить качество как основную цель улучшения общего качества программного обеспечения. Целевые показатели качества должны быть определены и задокументированны с участием команды проекта и клиента. Это гарантирует, что цели и качество станут достижимыми и измеримыми. MSF определяет качественные цели проекта в начале и подчеркивает выполнение этих целей в качестве основной части проекта.

Непрерывная проверка качества – это набор процедур, которые документируют каждое изменение в ходе проекта, необходимое, чтобы окончательно гарантировать качество. Не только отчеты о состоянии проекта, но и оценки текущей деятельности и возможных изменений необходимых для выявления проблем, как можно скорее. Для поддержки этих процедур, каждый проект нуждается в определенном процессе управляемых изменений. Все эти действия могут быть реализованы в виде поддерживающих действий рабочих процессов. Непрерывная проверки качества включает в себя тщательное тестирование. Кроме внутреннего тестирования, внешние приемочные испытания с клиентом необходимы также для того, чтобы убедиться, что продукт отвечает потребностям и требованиям заказчика. Поэтому процесс разработки программного обеспечения должен включать в себя рабочий процесс тестирования на протяжении всего процесса, в том числе внешних испытаний с конечными пользователями, чтобы обеспечить высокое качество программного обеспечения.

Требования заказчика – процесс разработки программного обеспечения основывается на четкой структуре и методологии требовать с заказчика документ. Он также должен интегрировать эти требования в полном процессе. Выявление требований является одним из наиболее сложных программных обеспечений инженерных дисциплин. Потребности и пожелания клиента, которые

обычно не имеют глубоких технических знаний, должны быть документированы, так чтобы разработчики могли создавать приложения на основе этой информации. Таким образом, необходимо, чтобы команда проекта понимала клиента и его бизнес. В противном случае будет невозможно правильно реализовать потребности клиентов. Процесс разработки программного обеспечения должен сосредоточиться на требованиях на протяжении всего проекта. Выявление и документирование требований в начале не является достаточным. Кроме того, процесс разработки программного обеспечения должен гарантировать, что не только клиент, но и конечные пользователи участвуют в процессе формирования требований. Успех проекта сильно зависит от этих двух групп: первая покупка продукта, а вторая, его использование. Процесс разработки программного обеспечения должен определить процедуры для подготовки конечных пользователей использовать конечный продукт.

Архитектура системы – в современных разработках программного обеспечения, архитектура системы оказывает значительное влияние на общее качество продукта. Одной из причин этого является интеграция в существующие системы программной среды, как большая часть сегодняшнего развития программного обеспечения. Повторное использование приобретает все большее значение в связи с увеличением времени и давления на затраты.

Фокус на команде – команда должна рассматриваться как совокупность равноправных лиц, которые вместе несут ответственность за качество и успешность проекта. Когда ответственность за неудачи может быть отнесена к одному человеку, успех проекта не может быть гарантирован. Сосредоточение на совместную работу повышает мотивацию участников проекта, так как все это рассматривается как столь же важной частью проекта. Это в конечном итоге приводит к высокой идентификации членов команды с продуктом. Очевидно, что мотивированные члены команды способствуют высокому качеству, поскольку они работают более концентрированно и добросовестно. Процесс разработки программного обеспечения должен включать четко определенную структуру команды, в том числе эффективной постановки задач и четких руководящих принципов коммуникации. MSF присваивает значение равного по каждому члену команды и роли: полная команда имеет шесть целей для достижения и может

рассматриваться как мощная реализация принципа фокуса команды. В большинстве проектов некоторые цели имеют более высокое значение, чем другие. Менеджеры решают самостоятельно или возлагают ответственность за неспособность членов своей команды. MSF «команда коллег» концепция позволяет избежать этих проблем и позволяет проекту получить прибыль от всех возможностей коллективной работы.

Парное программирование – парное программирование тесно связано с акцентом на команды, но было выбрано в качестве другого критерия оценки, поскольку оно было недооценено за свой вклад в высокое качество в прошлом. XP демонстрирует, как два разработчика могут дополнять друг друга, а не мешать друг другу. Один разработчик реализует текущий метод, а другой работает над вопросами интеграции. Такой подход экономит время и минимизирует количество ошибок. Лучшее решение, более вероятно, так как два человека, скорее всего, имеют разные точки зрения той же проблемы и, следовательно, дополняют друг друга в ее решении.

Управление требованиями – процесс разработки программного обеспечения должен быть определен и формализован на пути относительно его применения к широкому набору проектов. Процесс, который концентрируется на небольшом или специализированном наборе проектов гарантирует высокое качество в конкретной среде. Основной целью процесса является применение многих проектов внутри компании, не снижая его прочности. Процесс должен быть адаптирован к различным проектам на основе ее основных элементов. Для того, чтобы охватить всю сложность типовых проектов, заданный набор основных элементов должен быть сохранен. Большое количество основных элементов, не обязательно улучшат качество конечного продукта. Поэтому процессы разработки программного обеспечения должны опираться на основные элементы. Основываясь на этих основных элементах, необходимо определить методы интегрирования процесса к типу проекта и размер проекта.

Конфигурация и управление изменениями – хорошо функционирующая конфигурация и управление изменениями (КЗУ) является важной частью обеспечения качества программного обеспечения. Существование или не существование КЗУ оказывает наибольшее влияние при обслуживании программного

обеспечения и развития, все же основы для КЗУ (надлежащей документации, кода архивов источника и т.д.) должны быть установлены в процессе разработки.

Управление рисками – это процесс принятия и выполнения управленческих решений, направленных на снижение вероятности возникновения неблагоприятного результата и минимизацию возможных потерь, вызванных его реализацией. В рамках управления рисками осуществляется количественная и качественная оценка вероятности достижения предполагаемого результата, неудачи и отклонения от цели.. MSF представляет свою собственную модель управления рисками. Она, следовательно, делает акцент на дисциплине, придавая этому важное значение для поддержания проекта на пути улучшения общего качества разработки. Управление рисками, как собственная дисциплина, является критерием для любого процесса разработки программного обеспечения с акцентом на улучшение качества программного продукта. Управление рисками в рамках встреч и других дисциплин предполагает, что идентификация и управление рисками становится менее важным, поскольку проектная группа занята другими проблемами. Проект становится восприимчивым к факторам риска.

Таблица № 2.1

Критерии для выбора стандарта

Критерий	**MSF**	**RUP**	**XP**
Итерация	+	+	+
Качество как цель	+	-	-
Непрерывная проверка качества	+	+	+
Требования заказчика	+	+	+
Архитектура системы	+	+	+
Фокус на команде	+	+	+
Парное программирование	-	-	+
Управление требованиями	+	+	-
Конфигурация и управление изменениями	-	+	-
Управление рисками	+	-	-

.

Microsoft Solutions Framework является наиболее сбалансированной технологией, ориентированной на проектные группы малых и средних размеров. MSF не накладывает никаких ограничений на используемый инструментарий и содержит рекомендации весьма общего характера. Однако, эти рекомендации могут быть использованы для построения конкретного процесса, соответствующего потребностям коллектива разработчиков. [10]

Наш проект является небольшим, включает в себя 2 человека и этапы разработки и тестирования проводятся в среде разработки C#. Кроме того, основным преимуществом MSF является итерационная модель одновременно с уточняющими вехами (аналог каскадной модели). Таким образом, реализация MSF попыталась объединить каскадную и итерационную модель разработки и внедрения ПО.

По описанным выше преимуществам, мною был выбран стандарт MSF как наиболее гибкий и удобный для реализации моего проекта.

Одним из преимуществ этого стандарта является возможность управлять одновременно и проектом разработкой приложения, а также внедрением инфраструктуры.

Этапы

Предвидение, фаза процесса MSF начинается с фазы выработки концепции. Предвиденье может быть определено как создание широкого описания целей и ограничений проекта. На этом этапе определена команда и то, что команда должна выполнить для клиента. Цель фазы выработки концепции заключается в создании общей концепции проекта среди всех ключевых заинтересованных сторон проекта.

Методология разработки во время фазы выработки концепции команда управления программой определяет задачи и ожидаемые результаты, которые учитывают потребности и цели проекта. Эта фаза завершается видением. Этот этап означает, что клиент и команда договариваются о цели и направлении проекта.

Фаза планирования, на этапе планирования, команда определяет, что нужно для разработки плана, как создать решение. Команда готовит

функциональную спецификацию, создает дизайн решения, и готовит планы работы, сметы расходов и графики для различных результатов.

Этап планирования включает анализ требований. Эти требования могут быть классифицированы как бизнес-требования, требования пользователей, эксплуатационные требования и требования к системе. Эти требования используются для разработки решения и в особенности для проверки правильности конструкции.

После сбора и анализа требований, команда создает дизайн решения. Команда создает профили пользователей, которые определяют различные пользовательские решения их роли и обязанности. Затем команда создает ряд сценариев использования. Сценарий использования определяет деятельность, выполняемую определенным типом пользователя. Таким образом, команда должна создать сценарии использования для всех профилей пользователей. После создания сценариев использования, команда создает случаи использования для сценариев использования. Прецедент определяет последовательность шагов, которые пользователь будет выполнять в сценарии использования.

Разработка фазы во время фазы разработки, команда проекта создает решение. Этот процесс включает в себя создание кода, который реализует решения и документирование кода. В дополнение к разработке кода, команда также развивает инфраструктуру для решения.

В процессе разработки команда выполняет следующие основные задачи на данном этапе:

- Начало цикла разработки. Проверка того, что все задачи, выявленные в ходе выработки концепции и планирования этапов были завершены, так что команда может приступить к разработке решения.

- Создание приложения прототипа. Проверка концепций проекта решения в среде, которая напоминает среду, в которой решение будет в конечном счете развернуто. Эта среда является как можно более близкой к производственной среде. Эта задача будет завершена до начала разработки.

- Разработка компонентов решения. Разработка основных компонентов решения, и расширение этих компонентов к конкретным потребностям решения.

- Построение решения. Серия ежедневных и частых тестов, которые достигают высшей точки когда команда разработчиков выявляет ключевые особенности решения.

- Закрытие фазы разработки. Завершение всех функций, а также поставка кода и документации. Решение считается завершенным, и команда входит в процесс утверждения проекта.

Методология разработки

Стабилизирующая фаза, во время фазы стабилизации, команда выполняет интеграцию, нагрузку, и бета-тестирование решения. Кроме того, команда тестирует сценарии развертывания для решения. Команда сосредоточена на выявлении, приоритизации в решении вопросов, с тем, что решение может быть подготовлено к выпуску. На этом этапе решение переходит от состояния всех функций, являющихся полными, как это определено в функциональной спецификации для этой версии в состоянии удовлетворения определенных уровней качества. Кроме того, решение готово к развертыванию в бизнесе.

Результатами стабилизирующей фазы заключаются в следующем:

- окончательный релиз
- заметки о выпуске
- поддержка производительности элементов
- результаты испытаний и инструменты тестирования
- исходный код и исполняемые файлы
- проектные документы
- обзор решения

Развертывание фазы, на этом этапе команда развертывает технологию решения и компоненты сайта, стабилизирует развертывание, передает проект на операцию и поддержку, и получает окончательное одобрение от клиента по проекту. После развертывания команда проводит обзор проекта и исследование удовлетворенности клиентов. Фаза развертывания достигает кульминации в развертывании полной вехи.

Под моделью жизненного цикла понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении жизненного цикла. Модель жизненного цикла

зависит от специфики информационной системы и специфики условий, в которых последняя создается и функционирует. [11]

К настоящему времени наибольшее распространение получили следующие основные модели жизненного цикла:

- Модель водопада
- V-образная модель
- Модель эволюционного прототипирования
- Спиральный метод
- Итеративный и инкрементальный метод

Модель водопада на рис. 2.1 представляет собой линейный последовательный поток. В котором прогресс рассматривается как непрерывно нисходящий (как водопад) через фазы реализации программного обеспечения. Это означает, что любой этап в процессе разработки начинается, только если предыдущий этап завершен. Водопадный подход не определяет процесс возврата к предыдущему этапу для обработки изменений в требованиях. Водопадный подход является самым ранним и наиболее широко известным подходом, который использовался для разработки программного обеспечения.



Рисунок 2.1 – Водопадная модель

Проекты, которые не ориентированы на изменение требований, например, проекты, инициированные по запросу предложений когда у клиента есть ряд очень четко задокументированных требований.

Таблица № 2.2

Преимущество и недостатки водопадной модели

Преимущества	**Недостатки**
▪ *Легко объяснить пользователям.* ▪ *Структурный подход.* ▪ *Этапы и мероприятия четко определены.* ▪ *Помогает планировать и планировать проект.* ▪ *Проверка на каждом этапе обеспечивает раннее обнаружение ошибок / недоразумений.* ▪ *Каждый этап имеет конкретные результаты.*	▪ Предполагается, что требования системы могут быть заморожены. ▪ Очень сложно вернуться на какой-либо этап после его завершения. ▪ Немного гибкости и настройки объема сложно и дорого. ▪ Дорого и требует больше времени, помимо подробного плана.

V-образная модель на рис. 2.2 это расширение модели водопада. Вместо линейного перемещения ступени процесса после фазы реализации и кодирования изгибаются вверх, чтобы сформировать типичную V-образную форму. Основное различие между V-образной моделью и моделью водопада заключается в раннем планировании испытаний в V-образной модели. Требования к программному обеспечению четко определены и известны. Технологии и инструменты разработки программного обеспечения хорошо известны.

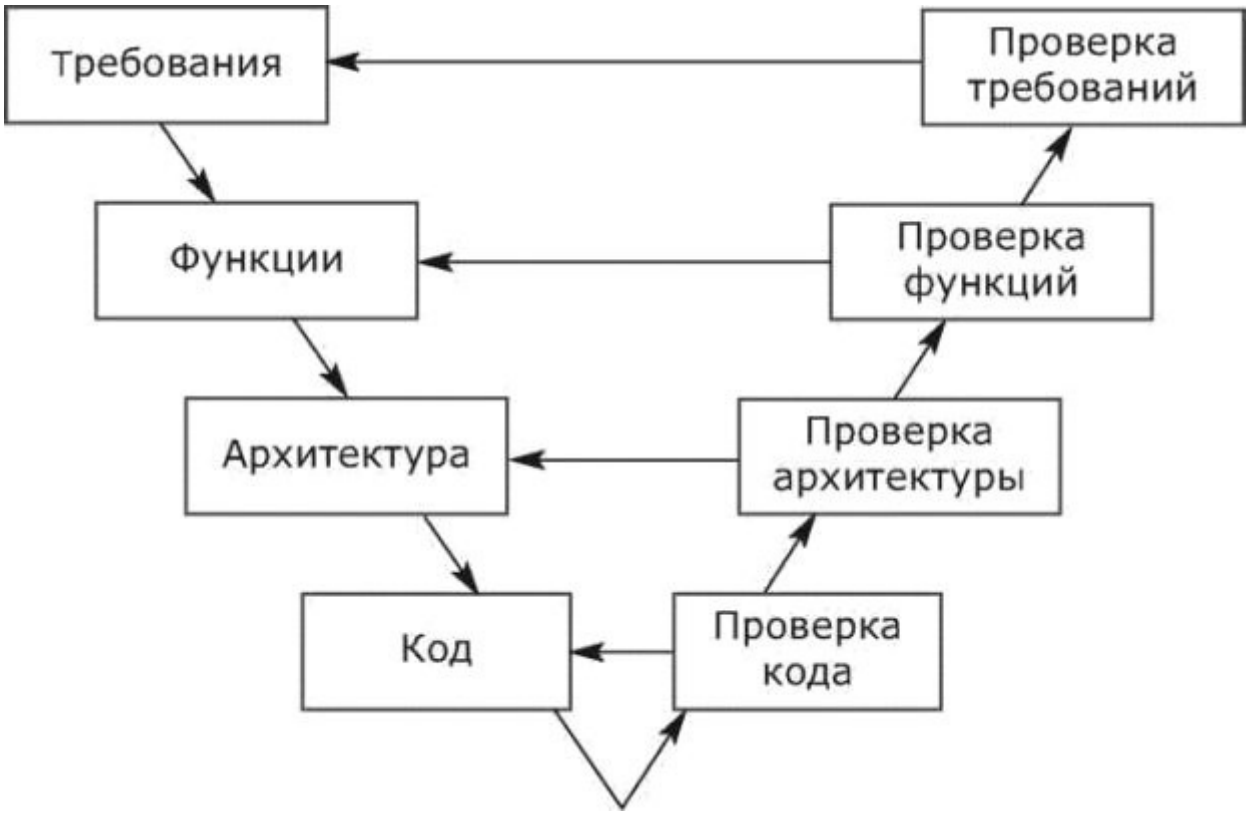


Рисунок 2.2 – V-образная модель

Таблица № 2.3

Преимущества и недостатки V-образной модели

Преимущества	**Недостатки**
▪ *Простой и удобный в использовании* ▪ *Каждый этап имеет конкретные результаты.* ▪ *Более высокие шансы на успех по модели водопада из-за разработки планов испытаний на ранних этапах жизненного цикла.* ▪ *Хорошо работает там, где требования легко понять.* ▪ *Проверка и валидация продукта на ранних этапах разработки продукта.*	▪ Очень не гибкий, как модель водопада. ▪ Регулировка объема сложна и дорога. ▪ Программное обеспечение разрабатывается на этапе реализации, поэтому ранние прототипы программного обеспечения не производятся. ▪ Модель не дает четкого пути для проблем, обнаруженных на этапах тестирования. ▪ Дорого и требует больше времени, в дополнение к детальному плану

Модель прототипирования на рис. 2.3 относится к деятельности по созданию прототипов программных приложений, например, неполных версий разрабатываемой программы. Это действие, которое может происходить при разработке программного обеспечения, и оно используется для визуализации некоторого компонента программного обеспечения, чтобы ограничить разрыв в неправильном понимании требований заказчика командой разработчиков. Это также уменьшит количество итераций, которые могут возникнуть в подходе с водопадом, и их трудно реализовать из-за негибкости подхода с водопадом. Таким образом, при разработке окончательного прототипа требование считается замороженным.

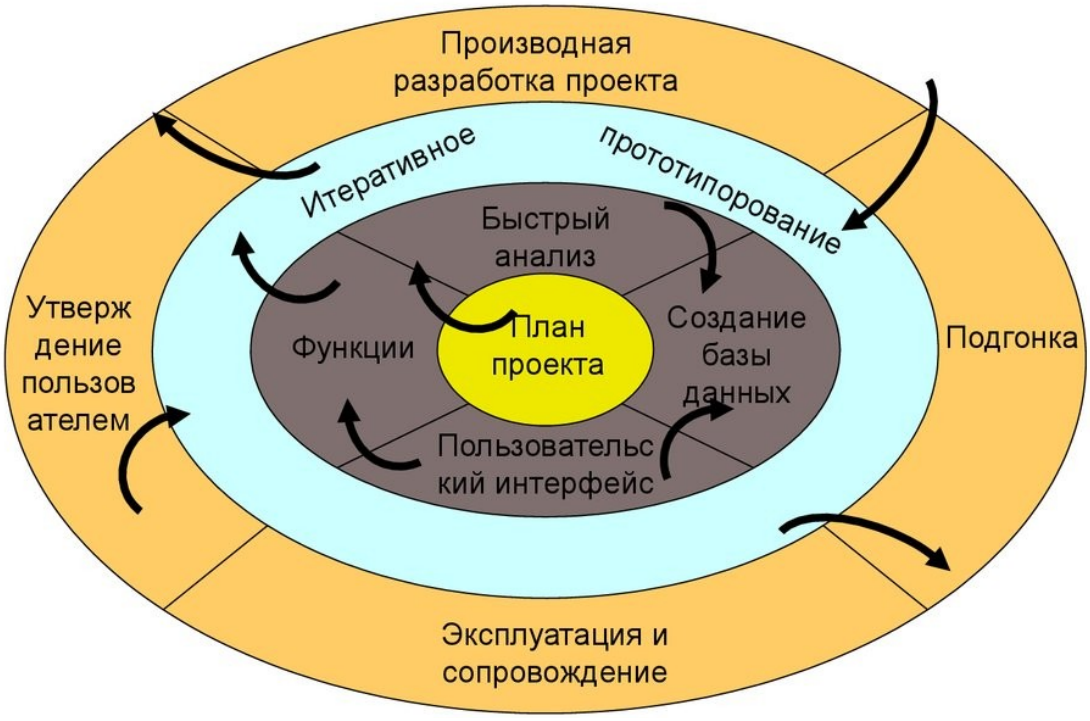


Рисунок 2.3 – Модель прототипирования

Таблица № 2.4

Преимущества и недостатки модели прототипирования

Преимущества	**Недостатки**
▪ *Сокращение времени и затрат, но это может быть недостатком, если разработчик теряет время на разработку прототипов.* ▪ *Улучшено и увеличено участие пользователей.*	▪ Недостаточный анализ. Путаница пользователя с прототипом и готовой системой. ▪ Недопонимание разработчиком целей пользователя. ▪ Избыточное время разработки прототипа. ▪ Это дорого для реализации прототипов

Спиральная модель (SDM) на рис. 2.4 она объединяет элементы как проектирования, так и поэтапного создания прототипа, чтобы объединить преимущества концепций сверху вниз и снизу вверх. Эта модель развития сочетает в себе особенности модели прототипа и модели водопада. Спиральная модель предпочтительна для больших, дорогих и сложных проектов. Эта модель использует многие из тех же этапов, что и модель водопада, в основном в том же порядке, разделенных планированием, оценкой риска и созданием прототипов и симуляций.

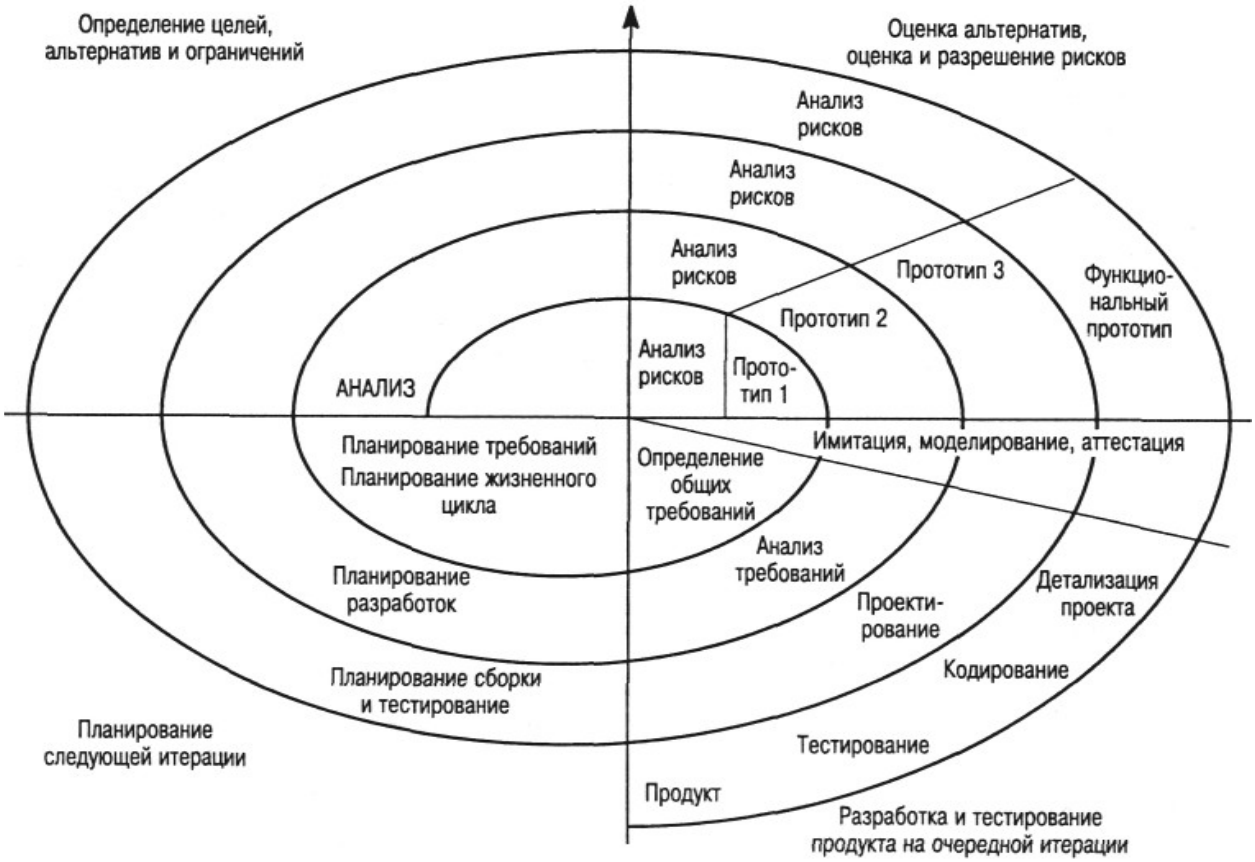


Рисунок 2.4 – Спиральная модель

Он используется в крупных приложениях и системах, которые встроены в небольшие фазы или сегменты.

Таблица № 2.5

Преимущества и недостатки спиральной модели

Преимущества	**Недостатки**
▪ *Оценки (т. Е. Бюджет, график и т. Д.) Становятся более реалистичными по мере продвижения работы, поскольку важные проблемы обнаруживаются ранее.* ▪ *Раннее привлечение разработчиков.* ▪ *Управляет рисками и развивает систему в несколько этапов.*	▪ Высокая стоимость и время достижения конечного продукта. ▪ Требуются специальные навыки для оценки рисков и предположений. ▪ Высоко настраиваемое ограничение повторного

использования

Итерационная и инкрементная модель на рис. 2.5 она разработана для преодоления слабых сторон модели водопада. Она начинается с первоначального планирования и заканчивается развертыванием с циклическими взаимодействиями между ними. Основная идея этого метода состоит в том, чтобы разработать систему с помощью повторяющихся циклов (итеративно) и меньшими порциями за один раз (постепенно), позволяя разработчикам программного обеспечения использовать преимущества того, что было изучено при разработке более ранних частей или версий системы. Может состоять из мини-водопадов или мини-V-образной модели.



Рисунок 2.5 - Итерационная и инкрементная модель

Он используется в больших системах, в которые встроены небольшие фазы или сегменты. Также может использоваться в системе отдельные компоненты, например, система ERP. Например, мы можем начать с модуля бюджета в качестве первой итерации, а затем мы можем начать с модуля инвентаризации и так далее.

Таблица № 2.6

Преимущества и недостатки итерационной модели

Преимущества	**Недостатки**
▪ *Производит бизнес-ценность в начале жизненного цикла разработки.* ▪ *Лучшее использование ограниченных ресурсов благодаря правильному определению приращения.* ▪ *Может удовлетворить некоторые запросы на изменение между приращениями.* ▪ *Больше внимания уделяется потребительской ценности, чем линейным подходам.* ▪ *Мы можем обнаружить проблемы проекта и изменения ранее.*	▪ Требуется тяжелая документация. ▪ Выполняет определенный набор процессов. ▪ Определяет приращения на основе функциональных и функциональных зависимостей. ▪ Требует большей вовлеченности клиентов, чем линейные подходы. ▪ Разделение функций и функций может быть проблематичным. ▪ Интеграция между итерациями может быть проблемой, если она не учитывается при разработке и планировании проекта.

Для преодоления перечисленных проблем была выбрана спиральная модель жизненного цикла.

2.1.2 Ожидаемые риски на этапах жизненного цикла и их описание

Сам стандарт MSF даёт некую гарантию минимизации рисков, так как весь ЖЦ проекта разделён на этапы, на каждом этапе есть роли, за которыми

закреплены цели, которые должны быть достигнуты и всё же на каждой фазе есть некоторые риски:

В фазе выработки концепции могут возникнуть следующие риски: недальновидный анализ сроков проекта и его бюджета.

Для ликвидации такого рода риска нужно более детально прорабатывать задачи и цели проекта, ставить больше контрольных точек.

Неправильно подобранный проектный состав исполнителей может повлечь полное отсутствие командной работы.

Данный риск уменьшается более тщательным подбором специалистов в проектную группу тестированием не только профессиональных навыков но и личностных качеств.

На фазе планирования могут возникнуть следующие риски:

Неправильно или не совсем корректно сформированная архитектура выбираемого решения.

Возможность появления этого риска зависит от компетенции руководителя проекта, на котором лежит принятие решение о выборе архитектуры разрабатываемого решения.

В фазе разработки возможны следующие риски:

Неправильная интерпретация технического задания и как следствие неправильное программирование архитектуры и сдвиг сроков.

Минимизацией данного риска служит более чёткое написание технического задания, понятного программисту.

Еще одним немаловажным риском в данном проекте является отсутствие должной квалификации у программиста в том языке, на котором решено реализовывать программу клиент, которая будет распределять заявки между инженерами.

В случае, если программист не будет укладываться в заданные временные рамки календарного плана проекта, придется использовать внешнего разработчика, так называемый «аутсорсинг» или «фриланс».

В фазе тестирования могут возникнуть следующие риски:

Риски неоконченного тестирования.

Может произойти ситуация что программный продукт будет протестирован не до конца.

Решается путем повторного тестирования на следующей итерации разработки.

В фазе внедрения могут возникнуть следующие риски:

Риски неправильного принятия решения о законченности части проекта.

Возникновение данных рисков ведет за собой проблему незаконченности решения и возможность возникновения нестыковок с другими частями разрабатываемой ИС.

Устраняется путем доработки при следующей итерации. [12]

2.1.3. Организационно-правовые и программно-аппаратные средства обеспечения информационной безопасности и защиты информации

Для данного комплекса задач существует несколько реализаций информационной безопасности.

Защита от внутренних угроз. Подразумевает разграничение прав пользователей. Подробные права пользователей описаны в таблице 2.7

Таблица № 2.7

Разграничение прав пользователей.

Группы пользователей	**Создание заявки**	**Возможность редактирования своей заявки**	**Возможность переназначит заявку**	**Работа с базой знаний**
Программист	Нету	Есть	Нету	Чтение/ создание/ удаление
Администратор	Чтения	Есть	Есть	Чтение

В компании ООО «Техторг» используются все возможные методы защиты информации, так как нет уникального одного метода, который смог бы обеспечить полную информационную безопасность, а сочетание всех методов позволяет реализовать максимальную информационную безопасность.

Также предприятие использует простой идентификатор пользователя / пароля. Оказывается, эту однофакторную аутентификацию чрезвычайно легко взломать. Надлежащая политика паролей должна быть введена в действие, чтобы гарантировать, что пароли не могут быть скомпрометированы. Ниже приведены некоторая политика предприятия, которые организация применяет.

Требует сложные пароли. Одна из причин взлома паролей заключается в том, что их легко угадать. Недавнее исследование показало, что тремя основными паролями, которые люди использовали в 2012 году, были пароли, 123456 и 12345678 . [13] Пароль не должен быть простым словом, которое можно найти в словаре. В первую очередь хакер попытается взломать пароль, проверив каждый термин в словаре. Вместо этого, хорошая политика паролей - это политика, которая требует использования как минимум восьми символов и как минимум одной заглавной буквы, одного специального символа и одного числа.

Меняйте пароли регулярно. Важно, чтобы пользователи регулярно меняли свои пароли. Пользователи должны менять свои пароли каждые шестьдесят или девяносто дней, гарантируя, что любые пароли, которые могли быть украдены или угаданы, не смогут быть использованы против компании.

Обучать сотрудников не выдавать пароли. Один из основных методов, которые используются для кражи паролей, просто выяснить их, спросив пользователей или администраторов. Данный метод называется «Социальная инженерия», происходит, когда злоумышленник вызывает службу поддержки или администратора безопасности и притворяется, что он является конкретным авторизованным пользователем, испытывающим трудности при входе в систему. Затем, предоставив некоторую личную информацию об авторизованном пользователе, злоумышленник убеждает сотрудника по безопасности сбросить пароль и сказать ему новый. Другой способ, с помощью которого сотрудники могут обмануть пароли, - это «Фишинг» по электронной почте. Фишинг возникает, когда пользователь получает электронное письмо, которое выглядит так, как будто оно поступило из надежного источника, такого как его банк или работодатель. В электронном письме пользователю предлагается щелкнуть ссылку и войти на веб-сайт, который имитирует подлинный веб-сайт, и ввести свой идентификатор и пароль. [13]

2.2 Информационное обеспечение задачи

2.2.1 Информационная модель и её описание

На рис. 2.6 приведена информационная модель в виде схемы.

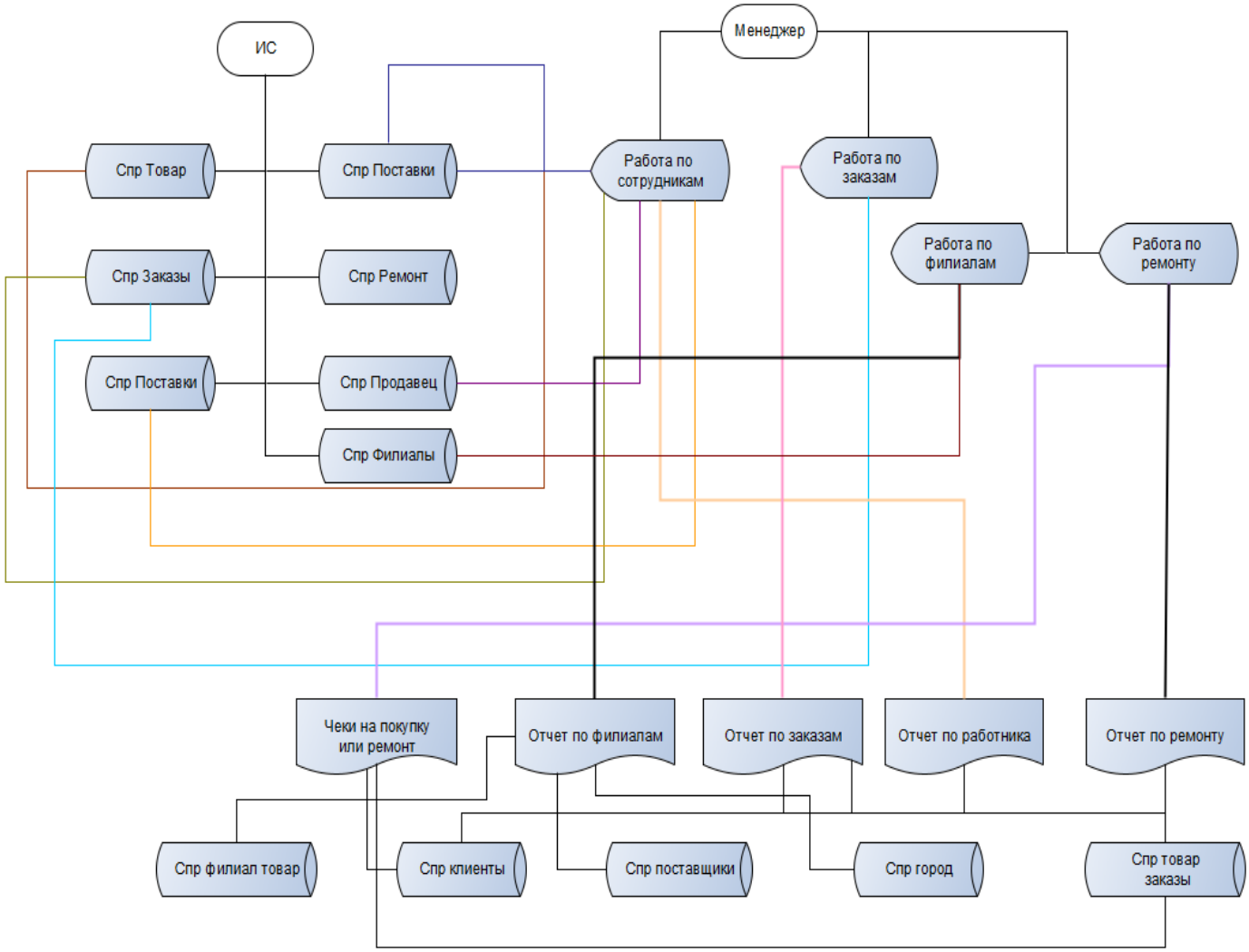


Рисунок 2.6 – Информационная модель

Данная информационная модель включает некоторые источники. Основной базой данных является созданная база данных Техторг. В ней находятся информация о работнике, информация о клиенте, информация о филиале, информация о заказе, информация по ремонту.

Для того чтобы войти в систему пользователь вводит логин и пароль, после логин и пароль проверяется в самой базе, затем осуществляется вход в систему. После чего пользователь имеет доступ к такому функционалу:

- просмотреть все заказы и вывести все на печать;
- просмотреть сотрудников и их работу;
- просмотреть работу по филиалам;
- просмотреть ремонт техники.

2.2.2 Характеристика нормативно-справочной, входной и оперативной информации

В данной выпускной работе используются только входящие справочники. Входящим документом является регистрация пользователя и просмотр документов по работе на предприятии. Регистрация пользователя представляет набор данных о клиенте. Просмотр документов предоставляется: по заказам и ремонтам. Примеры регистрации клиента и информации по заказам можно увидеть на таблицах 2.8 – 2.10.

Таблица № 2.8

Информация о клиенте

Имя	Olga
Фамилия	Kozlova
Отчество	Valeryevna
Номер телефона	(920) 295-3695
Адрес	428000 Cheboksary, Mira 7

Таблица № 2.9

Информация о заказе

Дата заказа	2019-07-29
Сумма	225096,00
Продавец	Aleksey T.

Таблица № 2.10

Информация состав заказа

Количество	3
Название товара	Dashboard camera

2.2.3 Характеристика результатной информации

В данной выпускной работе результирующей информацией являются – отчет о заказах. Пример приведен в таблице 2.11

Таблица № 2.11

Результирующая информация

Налоги	300 000.00 руб.
Зарплата	780 000.00 руб.
Чистый доход	1 500 000.00 руб.
Доход от заказов за месяц	140 000.00 руб.
Доход от заказов за год	1 700 000.00

2.3 Программное обеспечение задачи

2.3.1 Общие положения (дерево функций и сценарий диалога)

В данной выпускной работе автоматизируется документооборот предприятия, это та работа, которая вводится вручную. Сценарий просмотра, сколько сотрудник выполнил работы в этом месяце на рис. 2.7

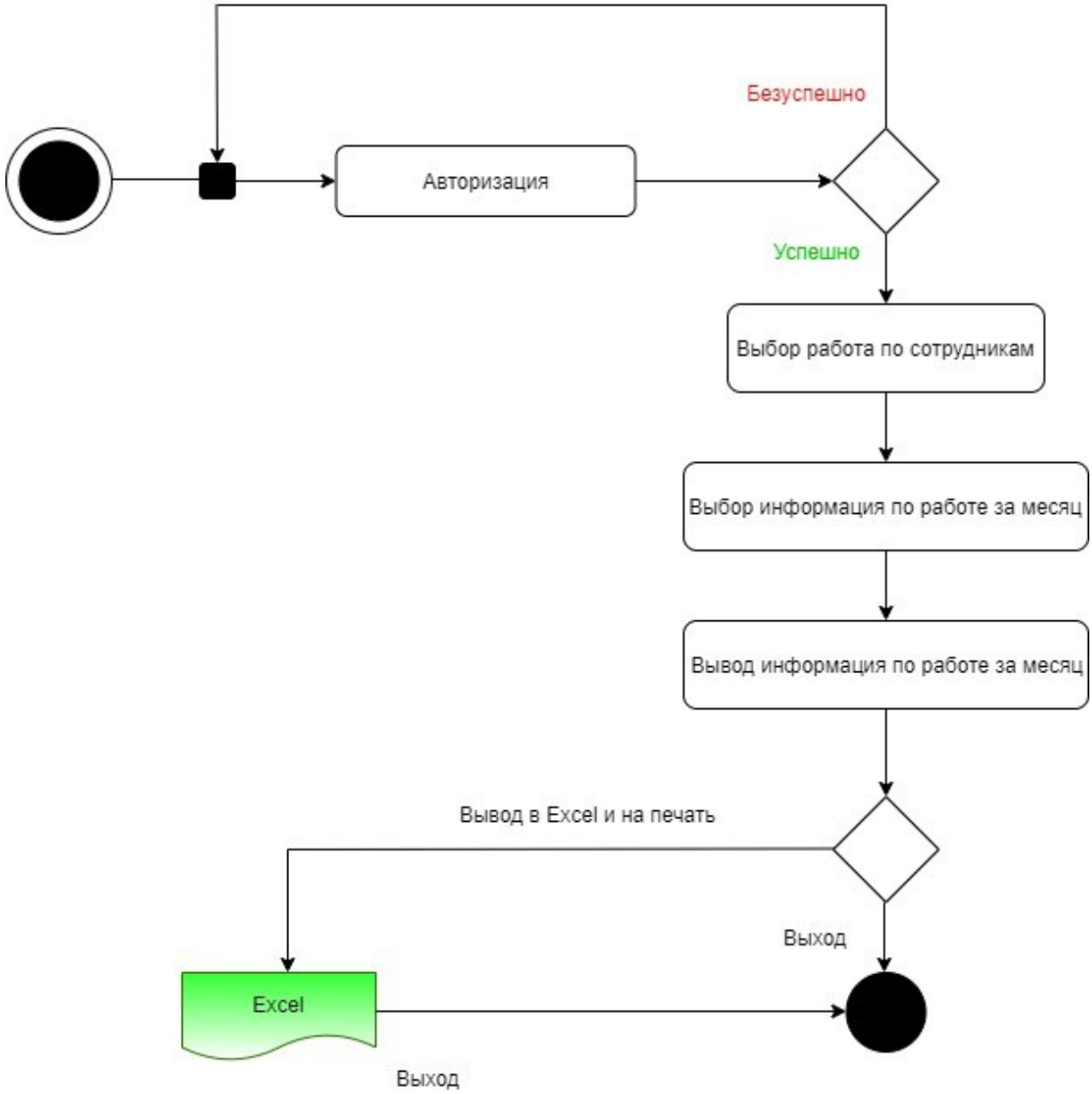


Рисунок 2.7 – сценария просмотра сотрудников

Программа согласно заложенной в неё логике обрабатывает всю информацию по заказам, зарплате и ремонтах.

Любой функционал программы можно разделить на основной, с помощью которой достигается основная цель алгоритма программы и дополнительный (служебный) это то, что можно настроить изменить, прояснить. Дерево функций как раз наглядно демонстрирует разделение данных функций. Дерево функций изображено на рисунке 2.8

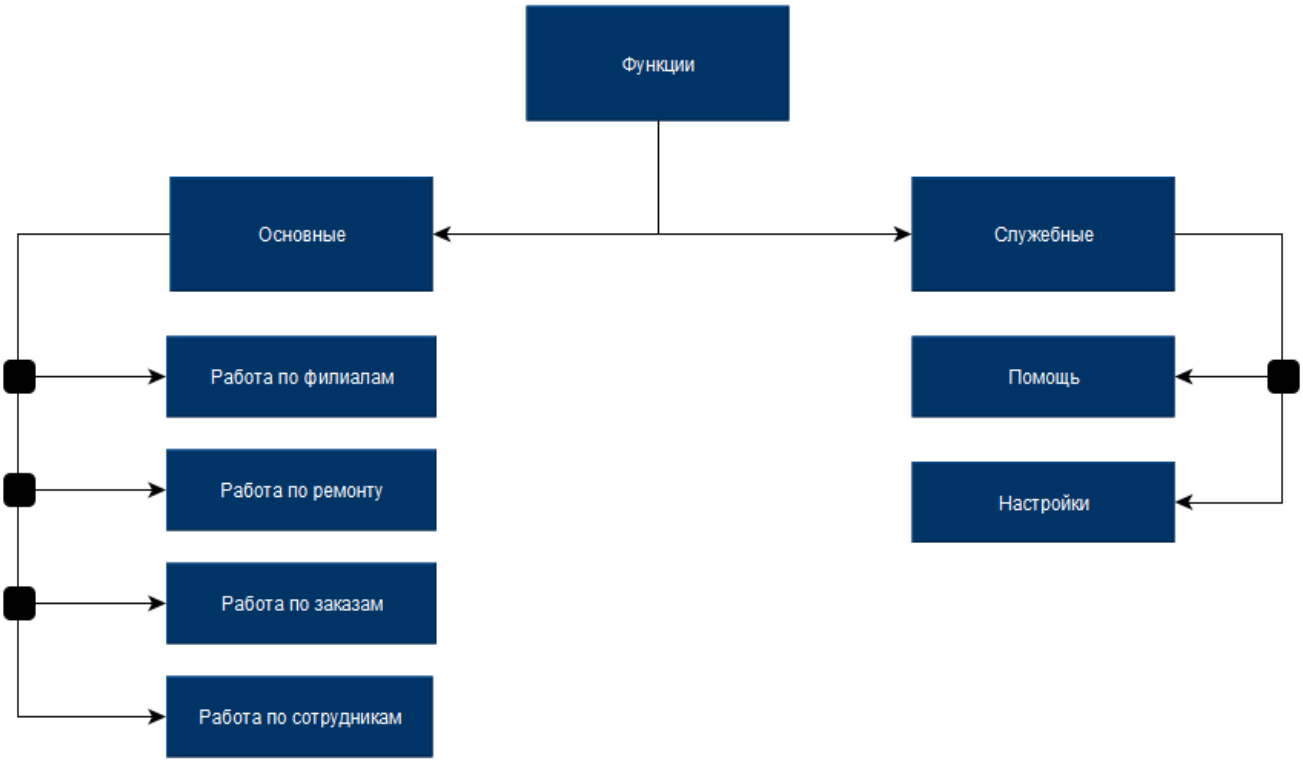


Рисунок 2.8 – Дерево функций

2.3.2. Характеристика базы данных

Разработка концептуальной модели предметной области.

В результате анализа требований пользователя и структур данных, описывающих деятельность, обработка документооборота, определены как сущности следующие объекты:

- Филиалы (Код_филиала – первичный ключ)
- Города (Код_города – первичный ключ)
- Клиенты (Код_клиента – первичный ключ)
- Поставщики (Код_поставщика – первичный ключ)
- Ремонт (Код_ремонта – первичный ключ)
- Работники (Код_работника – первичный ключ)

Первичный вариант концептуальной модели данных предметной области представляем в виде диаграммы «сущностей-связей» (ER-диаграммы), разработанной в программе yEd, показано на рисунке 2.16

При этом минимальная информация, изображаемая на диаграмме, должна содержать следующие пять элементов данных:

- сущности, изображаемые в виде прямоугольников произвольного размера;
- ключи сущностей, изображаемые в виде подчеркнутых надписей рядом с прямоугольником с любой стороны;
- связи, изображаемые в виде ромбов произвольного размера и соединенных прямыми линиями с соответствующими сущностями, эти линии могут быть произвольного наклона и, по возможности, без пересечений;
- степень участия в связи для каждой сущности, изображаемая в виде числа или буквы около соответствующего прямоугольника-сущности и линии связи;
- полнота участия в связи для каждой сущности, изображаемая в виде точки перед символом-степенью участия, при этом наличие точки означает полное участие в связи данной сущности, т.е. значение обязательный.

Каких-либо строгих правил для изображения диаграммы ER-типов не существует. В различных коммерческих системах (CASE средствах) проектирования БД используется и другая нотация для изображения ER-диаграмм, наиболее известными из них являются нотация «куриных лапок» и нотация стандарта IDEF1X. Однако, для обсуждения результатов проектирования с будущими пользователями БД удобнее всего представить диаграмму ER-типов в нотации, предложенной впервые П. Ченом, т.е. в виде прямоугольников и ромбов. [14]

В процессе документооборота сущности взаимодействуют друг с другом. В концептуальной модели взаимодействие между сущностями выражается с помощью связей [14], основными из которых являются следующие.

В связи Филиалы - Города (M: M) независимо от класса принадлежности сущностей формируются три отношения, два отношения соответствуют связываемым сущностям и их ключи являются первичными ключами этих отношений. Третье отношение является связным между первыми двумя, а его ключ объединяет ключевые атрибуты связываемых отношений. Связь изображена на рисунке 2.9

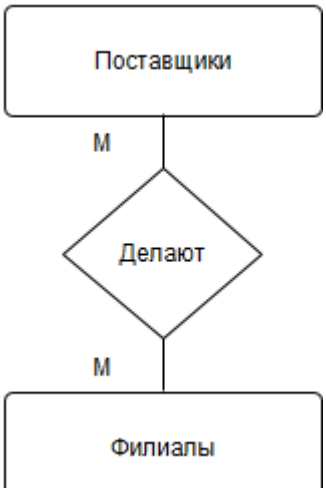


Рисунок 2.9 – Сущность связь (филиалы и города)

В связи Филиалы - Техника (M: M) независимо от класса принадлежности сущностей формируются три отношения, два отношения соответствуют связываемым сущностям и их ключи являются первичными ключами этих отношений. Третье отношение является связным между первыми двумя, а его ключ объединяет ключевые атрибуты связываемых отношений. Связь изображена на рисунке 2.10



Рисунок 2.10 – Сущность связь (филиалы и техника)

В связи Заказы - Техника (M: M) независимо от класса принадлежности сущностей формируются три отношения, два отношения соответствуют связываемым сущностям и их ключи являются первичными ключами этих отношений. Третье отношение является связным между первыми двумя, а его ключ объединяет ключевые атрибуты связываемых отношений. Связь изображена на рисунке 2.11

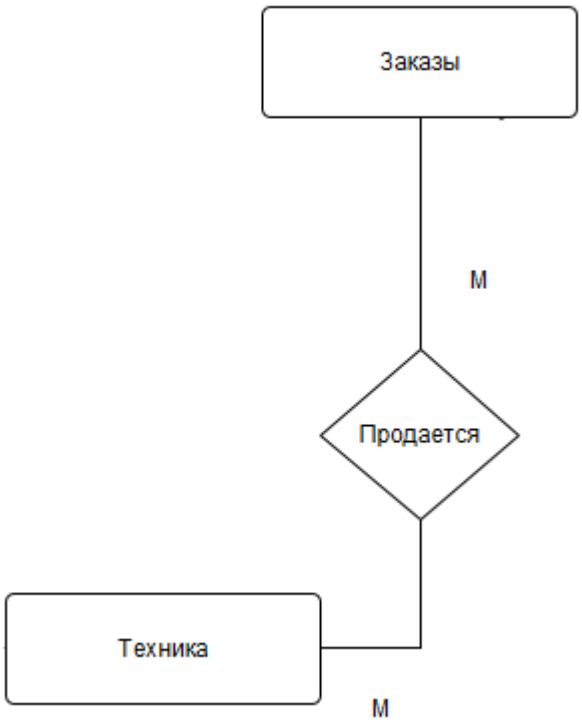


Рисунок 2.11 – Сущность связь (заказы и техника)

В связи Города - Филиалы (1: M) класс принадлежности сущности Филиалы будет обязательным. Согласно правилу 3, первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне M, и служит внешним ключом. И строится 2 таблицы.

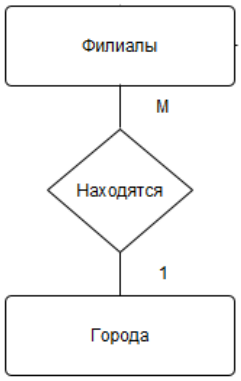


Рисунок 2.12 – Сущность связь (города и филиалы)

В связи Сотрудники - Ремонт (1: M) класс принадлежности сущности Ремонт будет обязательным. Согласно правилу 3, первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне M, и служит внешним ключом. И строится 2 таблицы.

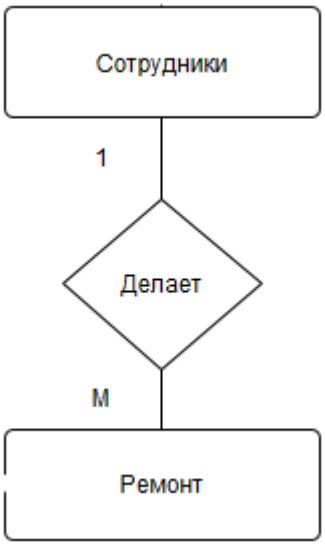


Рисунок 2.13 – Сущность связь (сотрудники и ремонт)

В связи Сотрудники - Ремонт (1: M) класс принадлежности сущности Ремонт будет обязательным. Согласно правилу 3, первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне M, и служит внешним ключом. И строится 2 таблицы.



Рисунок 2.14 – Сущность связь (клиенты и ремонт)

В связи Сотрудники - Заказы (1: M) класс принадлежности сущности заказы будет обязательным. Согласно правилу 3, первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне M, и служит внешним ключом. И строится 2 таблицы.

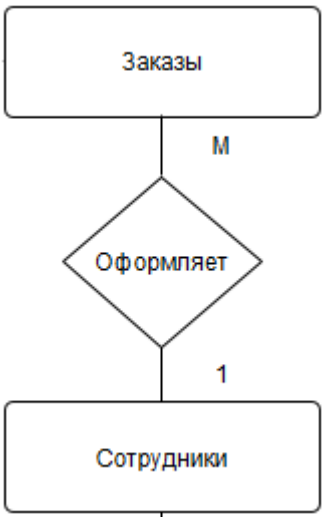


Рисунок 2.15 – Сущность связь (сотрудники и заказы)

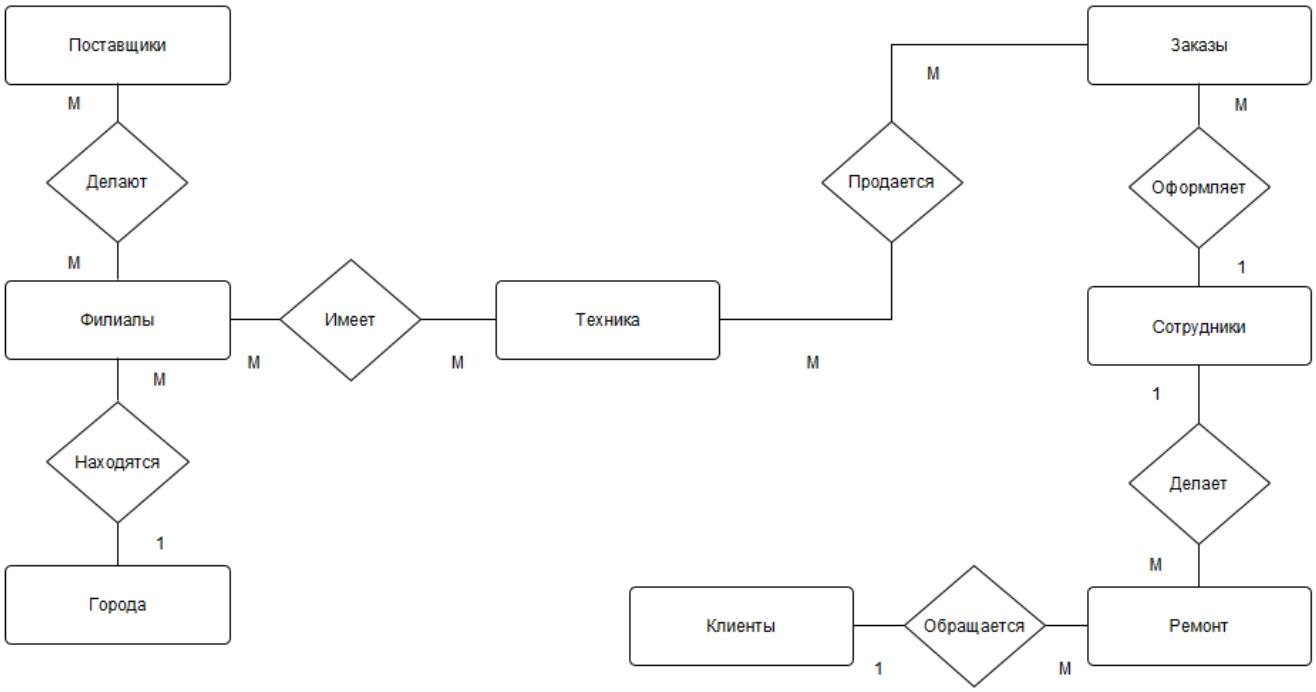


Рисунок 2.16 – ERD-модель

Логическая модель. Разработка логической модели базы данных.

После того, как концептуальная модель одобрена, можно перейти к этапу логического проектирования БД. При этом исходными данными для проектирования являются диаграмма ER-типов концептуальной модели предметной области и логическая модель данных, используемая для реализации будущей БД и поддерживаемая какой-либо коммерческой СУБД. [15]

В качестве логической модели данных выбираем реляционную модель когда, как наиболее распространенную в большинстве современных СУБД. [16] Теперь следует преобразовать концептуальную модель в соответствии с требованиями реляционной модели данных.

Таким образом, формально этап логического проектирования БД может в проекте отсутствовать, но при этом следует иметь в виду, что при выполнении преобразования диаграммы ER-типов в схему БД с помощью выше изложенных правил 1-8, мы, фактически и выполняем логическое проектирование БД с использованием реляционной логической модели данных. И в любом случае получаем логическую модель, в которой все объекты представлены как сущности (сильные или слабые) и связи один ко многим и один к одному (сильные или слабые).

Еще одним ограничением реляционной модели данных является требование нормализации всех сущностей-отношений. Это требование часто называют первой нормальной формой (1НФ). Оно заключается в том, чтобы все атрибуты в БД были атомарными с точки зрения СУБД. Атомарность означает неделимость каждого значения данных на многие значения, например не допускается иметь в БД атрибутов, представляющих собой списки значений (множественные значения), как-то: номера телефонов, оценки, дни работы и т.д. Не допускается иметь в БД и атрибуты составные, например адрес, включающий город, улицу, дом, квартиру при условии, что в процессе использования БД потребуется обращение к какой-либо части этого адреса.

Во всех этих случаях требуется на этапе логического проектирования обязательно провести нормализацию каждого отношения, т.е. привести его к 1НФ.

На следующем шаге проектирования необходимо оценить полученные отношения с точки зрения нормальных форм более высокого порядка. Причем, из пяти нормальных форм, известных в настоящее время 2НФ, 3НФ. НФБК, 4НФ и 5НФ, на практике, как правило, бывает вполне достаточно обеспечить третью нормальную форму или НФБК (усиленную 3НФ). В крайнем случае, можно вообще не заниматься вопросами приведения БД к той или иной НФ, но при этом надо помнить о том, что надежность проекта будет тем ниже, чем сложнее БД и чем большие требования предъявляются к целостности данных.

Чтобы провести проверку на уровень нормализации БД, необходимо:

- разместить все атрибуты во всех полученных сущностях-отношениях (слабых и сильных);
- выявить в каждом отношении потенциальные ключи;
- выявить все функциональные зависимости (ФЗ) между атрибутами в каждом отношении;
- сопоставить потенциальные ключи с детерминантами ФЗ (их левыми частями);
- если какой-либо детерминант отношения не совпадает ни с одним потенциальным ключом этого же отношения, то данное отношение не находится в той или иной НФ и его требуется нормализовать в соответствии с теорией нормальных форм.

В результате приведения БД в ту или иную НФ, как правило, появляются дополнительные отношения, которые следует добавить в логическую модель. [17]

Таким образом, получается окончательный вариант логической модели БД, ER диаграмма которой представлена на рисунке 2.17

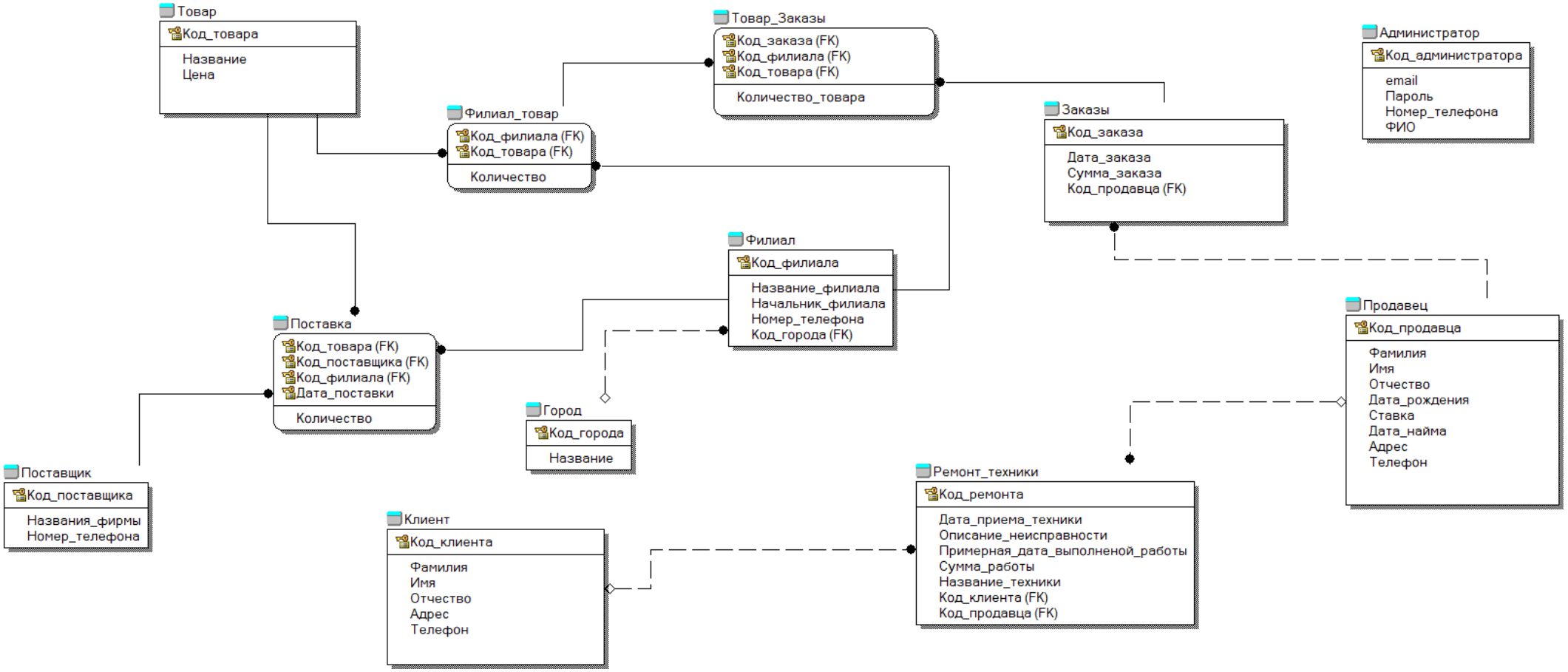


Рисунок 2.17 – Логическая модель

Разработка физической модели базы данных. На третьем этапе проекта, опираясь на логическую модель БД, полученную на предыдущем этапе.

На основе схемы базы данных, а именно, используя значения потенциальных и внешних ключей отношений, напишем скрипты создание таблиц.

Ниже приведен скрипт создания таблицы филиалы:

```
CREATETABLEBranches
(
BranchIdintIDENTITY (1,1) ,
Name nvarchar(50)  NULL ,
Director nvarchar(50)  NULL ,
PhoneDirectornvarchar(50)  NULL ,
CityIdint  NULL
)
ALTER TABLE Branches
ADDCONSTRAINTXPKФилиалPRIMARYKEY
```

Ниже приведен скрипт создания таблицы филиалы_продукты:

```
CREATE TABLE BranchesProducts
(
Quantity int  NULL ,
BranchIdint  NOT NULL ,
ProductIdint  NOT NULL
)
go
ALTER TABLE BranchesProducts
ADDCONSTRAINTXPKФилиал_товарPRIMARY
```

Ниже приведен скрипт создания таблицы города:

```
CREATE TABLE Cities
(
CityIdint IDENTITY (1,1) ,
Name nvarchar(50)  NULL
```

```
)
go
ALTER TABLE Cities
ADDCONSTRAINTXPKГородPRIMARYKEY
```

Ниже приведен скрипт создания таблицы клиент:

```
CREATE TABLE Customers
(
CustomerIdint IDENTITY (1,1) ,
Surname nchar(50)  NULL ,
Name nchar(50)  NULL ,
Patronymic nchar(50)  NULL ,
Address nchar(50)  NULL ,
Phone nchar(50)  NULL
)
go
ALTER TABLE Customers
ADDCONSTRAINTXPKКлиентPRIMARYKEY
```

Ниже приведен скрипт создания таблицы поставка:

```
CREATE TABLE Deliveries
(
DateOfDeliveryIddate  NOT NULL ,
Quantity int  NULL ,
ProductIdint  NOT NULL ,
ProviderIdint  NOT NULL ,
BranchIdint  NOT NULL
)
go
ALTER TABLE Deliveries
ADDCONSTRAINTXPKПоставкаPRIMARYKEY
```

Ниже приведен скрипт создания таблицы заказы:

```
CREATE TABLE Orders
(
OrderIdint IDENTITY (1,1) ,
DateOrdersdate  NULL ,
OrderPricemoney  NULL ,
SellerIdint  NOT NULL
)
go
ALTER TABLE Orders
ADDCONSTRAINTXPKЗаказыPRIMARYKEY
```

Ниже приведен скрипт создания таблицы продукты:

```
CREATE TABLE Products
(
ProductNamenvarchar(50)  NULL ,
Price money  NULL ,
ProductIdint IDENTITY (1,1)
)
go
ALTER TABLE Products
ADDCONSTRAINTXPKТоварPRIMARYKEY
```

Ниже приведен скрипт создания таблицы продукты_заказы:

```
CREATE TABLE ProductsOrders
(
Quantity int  NULL ,
OrderIdint  NOT NULL ,
BranchIdint  NOT NULL ,
ProductIdint  NOT NULL
)
go
```

```
ALTER TABLE ProductsOrders
ADD CONSTRAINT XPKТовар_Заказы PRIMARY KEY
```

Ниже приведен скрипт создания таблицы поставщик:

```
CREATE TABLE Providers
(
ProviderIdint IDENTITY (1,1) ,
CompanyNamenvarchar(50)  NULL ,
PhoneNumbernvarchar(50)  NULL
)
go
ALTER TABLE Providers
ADDCONSTRAINTXPKПоставщикPRIMARYKEY
```

Ниже приведен скрипт создания таблицы ремонт:

```
CREATE TABLE Repairs
(
RepairIdint IDENTITY (1,1) ,
OrderAcceptedDatedate  NULL ,
DescriptionOfMalfunctionnvarchar(50)  NULL ,
OrderAcceptancedate  NULL ,
OrderPricemoney  NULL ,
TechniqueNamenvarchar(50)  NULL ,
CustomerIdint  NULL ,
SellerIdint  NULL
)
```

```
ALTER TABLE Repairs
ADD CONSTRAINT XPKРемонт_техники PRIMARY KEY
```

Ниже приведен скрипт создания таблицы сотрудники:

```
CREATE TABLE Sellers
(
SellerIdint IDENTITY (1,1) ,
Surname nvarchar(50)  NULL ,
Name nvarchar(50)  NULL ,
Patronymic nvarchar(50)  NULL ,
DateOfBirthdate  NULL ,
Salary money  NULL ,
HiringDatedate  NULL ,
Address nvarchar(50)  NULL ,
Phone nvarchar(50)  NULL
)
go
ALTERTABLESellers
ADDCONSTRAINTXPKПродавецPRIMARYKEY
```

Ниже приведено создание связей между таблицами филиалы, филиалы_-продукты, поставка, заказы, продукты_заказы, ремонт.

```
ALTER TABLE Branches
ADD CONSTRAINT  R_20 FOREIGN KEY (CityId) REFERENCES Cities(CityId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
ALTER TABLE BranchesProducts
ADD CONSTRAINT  R_21 FOREIGN KEY (BranchId) REFERENCES Branches(BranchId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
```

```
ALTER TABLE BranchesProducts
ADD CONSTRAINT  R_22 FOREIGN KEY (ProductId) REFERENCES Products(ProductId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
ALTER TABLE Deliveries
ADD CONSTRAINT  R_24 FOREIGN KEY (ProductId) REFERENCES Products(ProductId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
ALTER TABLE Deliveries
ADD CONSTRAINT  R_25 FOREIGN KEY (ProviderId) REFERENCES Providers(ProviderId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
ALTER TABLE Deliveries
ADD CONSTRAINT  R_26 FOREIGN KEY (BranchId) REFERENCES Branches(BranchId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
ALTER TABLE Orders
ADD CONSTRAINT  R_15 FOREIGN KEY (SellerId) REFERENCES Sellers(SellerId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
ALTER TABLE ProductsOrders
```

ADD CONSTRAINT R_10 FOREIGN KEY (OrderId) REFERENCES Orders(OrderId)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE ProductsOrders

ADD CONSTRAINT R_23 FOREIGN KEY (BranchId,ProductId) REFERENCES BranchesProducts(BranchId,ProductId)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Repairs

ADD CONSTRAINT R_18 FOREIGN KEY (CustomerId) REFERENCES Customers(CustomerId)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Repairs

ADD CONSTRAINT R_19 FOREIGN KEY (SellerId) REFERENCES Sellers(SellerId)

ON DELETE NO ACTION

ON UPDATE NO ACTION

На основе написанных скриптов получаем схему связей между отношениями, изображено на рисунке 2.18

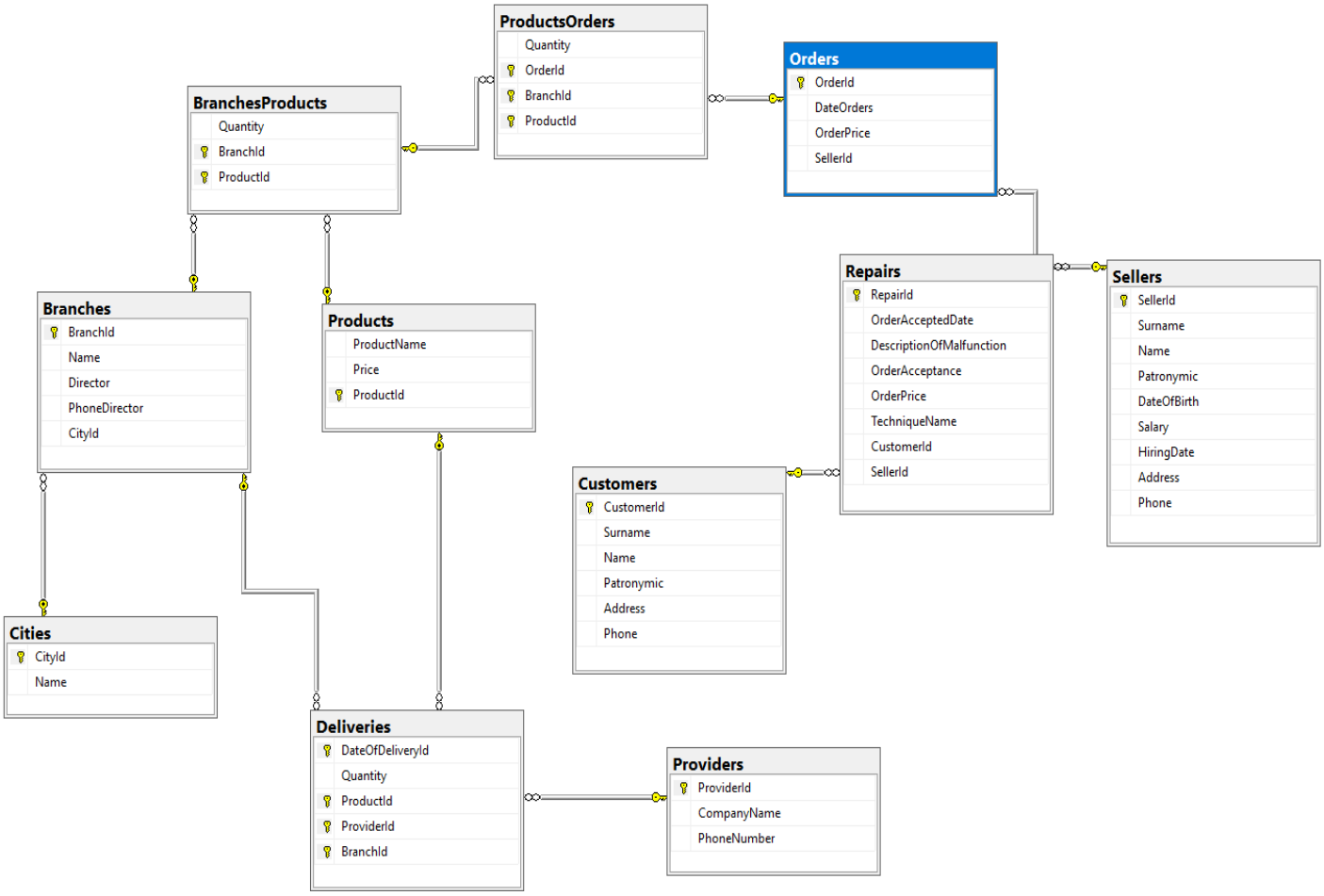


Рисунок 2.18 – Физическая модель

2.3.3 Структурная схема пакета (дерево вызова процедур и программ)

Схема графических форм разрабатываемой программы предоставлена на рисунке 2.19.

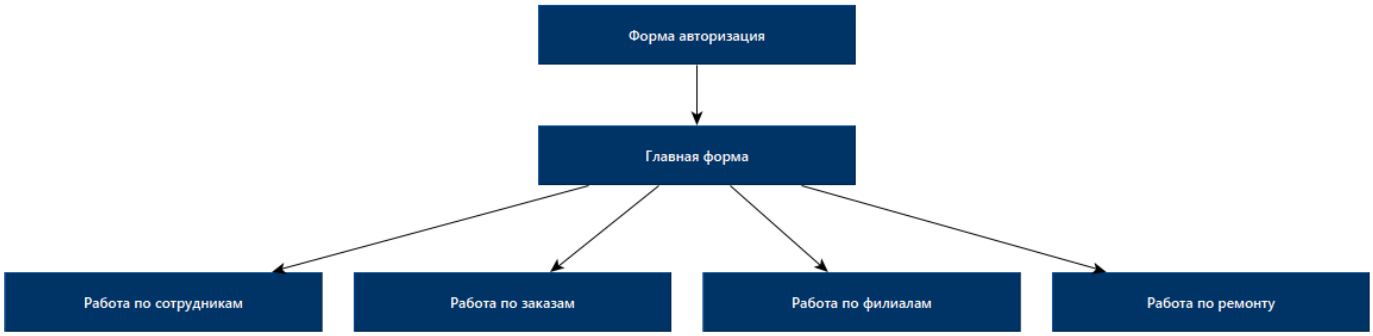


Рисунок 2.19 – Схема графических форм

В таблице 2.12 содержится описание всех модулей структурной схемы пакетов.

Таблица № 2.12

Описание всех модулей структурной схемы

Наименование модуля	**Функционал формы**
Форма «Авторизация»	Открывается при входе, идентифицирует пользователя и открывает соответствующую форму. Содержит ссылку на форму регистрации.
Форма «Главная»	Через главную форму осуществляется переход по следующим формам: работа по сотрудникам, работа по заказам, работа по филиалам, работа по ремонту.
Форма «Сотрудники»	Форма просмотра информации о сотрудниках на предприятии.
Форма «Заказы»	Форма просмотра информации по доходам и расходам на предприятии.
Форма «Филиалы»	Форма просмотра информации по филиалам.
Форма «Работа по ремонту»	Форма просмотра информации по заказанном ремонте.

2.3.4 Описание программных модулей

Для извлечения или отправки данных в базу данных была выбрана программная платформа «Entity Framework».

Entity Framework представляет специальную объектно-ориентированную технологию на базе Фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работает с объектами. [18]

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из бд, но и получать объекты, связанные различными ассоциативными связями.

Другим ключевым понятием является Entity Data Model. Эта модель сопоставляет классы сущностей с реальными таблицами в БД.

Entity Data Model состоит из трех уровней: концептуального, уровень хранилища и уровень сопоставления (маппинга). На концептуальном уровне происходит определение классов сущностей, используемых в приложении. Уровень хранилища определяет таблицы, столбцы, отношения между таблицами и типы данных, с которыми сопоставляется используемая база данных. Уровень сопоставления (маппинга) служит посредником между предыдущими двумя, определяя сопоставление между свойствами класса сущности и столбцами таблиц. Таким образом, мы можем через классы, определенные в приложении, взаимодействовать с таблицами из базы данных.

Entity Framework предполагает три возможных способа взаимодействия с базой данных:

- Database first: Entity Framework создает набор классов, которые отражают модель конкретной базы данных;
- Model first: сначала разработчик создает модель базы данных, по которой затем Entity Framework создает реальную базу данных на сервере;
- Code first: разработчик создает класс модели данных, которые будут храниться в бд, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы. [18]

По выше разработанной схеме БД были созданы соответствующие классы и объекты для работы с ними. Для описания работы программы наиболее понятным будет пояснение с помощью диаграммы классов.

Диаграммы классов предоставлена в виде схемы, которая отображена на

рисунке 2.20

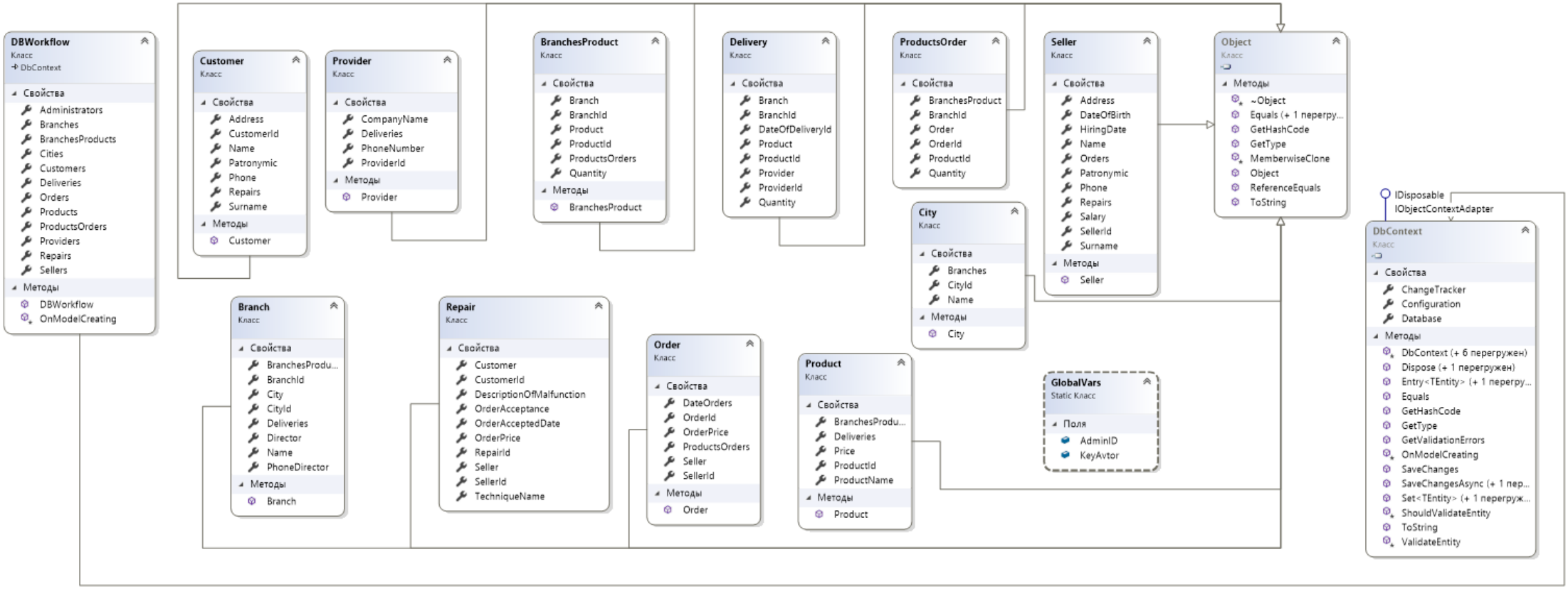


Рисунок 2.20 – Диаграмма классов

Далее представлены блок-схемы основных/расчетных модулей.

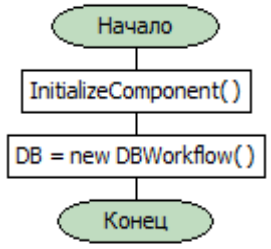


Рисунок 2.21 – Конструктор формы авторизации

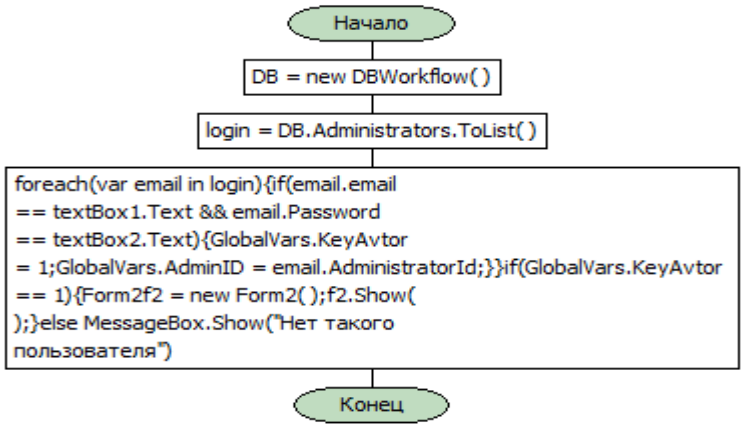


Рисунок 2.22 – Метод авторизации

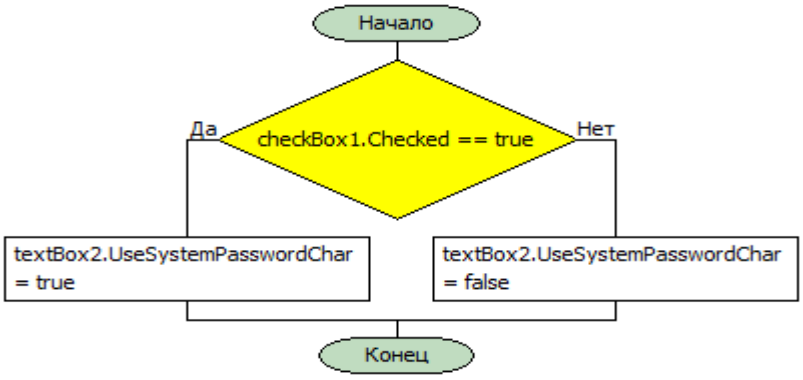


Рисунок 2.23 – Метод видимости пароля

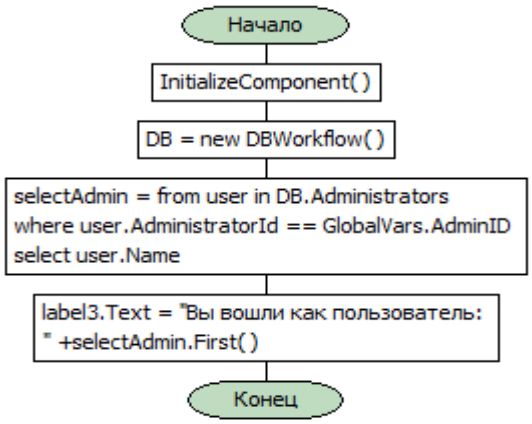


Рисунок 2.24 – Конструктор главной формы

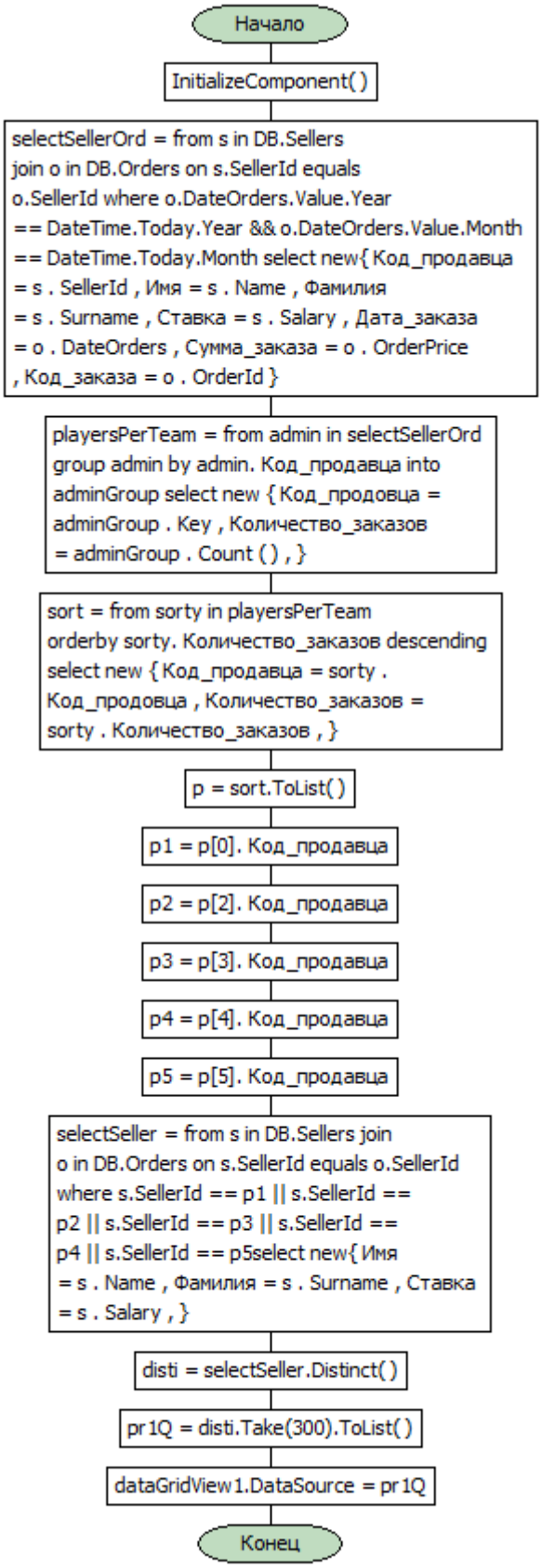


Рисунок 2.25 – Метод вывода топ сотрудников

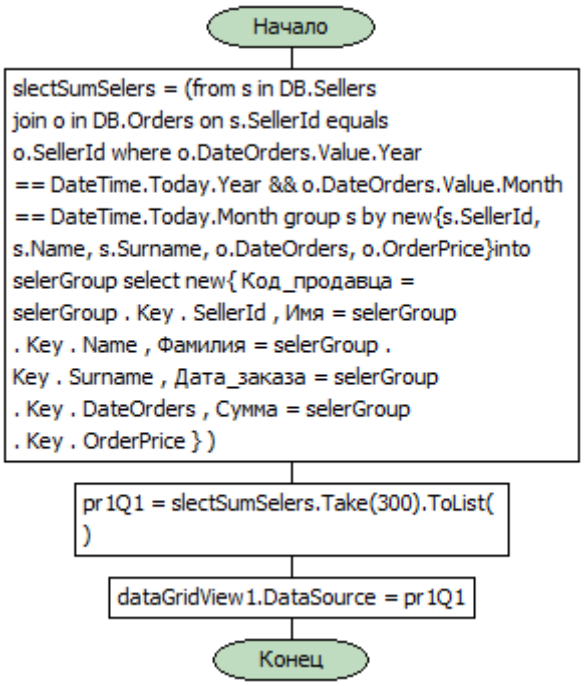


Рисунок 2.26 – Метод суммы заказов за месяц

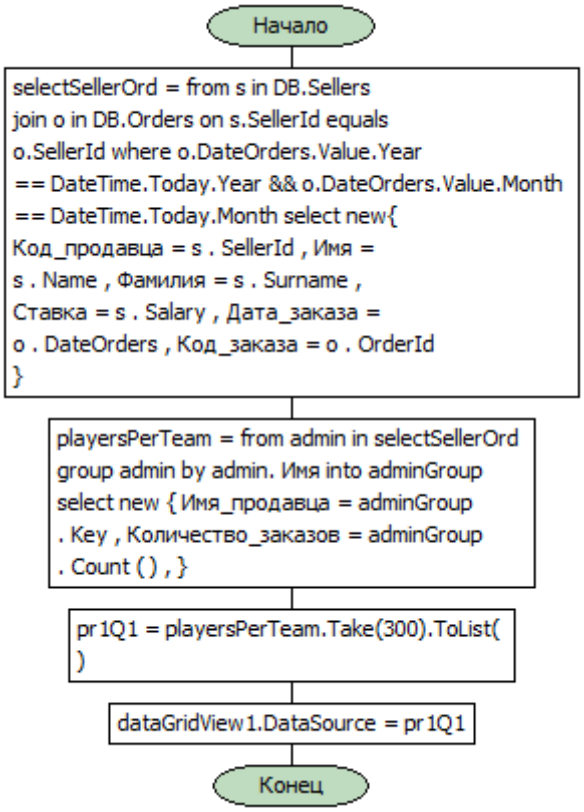


Рисунок 2.27 – Метод количество заказов за месяц

2.4 Контрольный пример реализации проекта и его описание

Для доступа к системе мы должны будем запустить exe-файл, ведущую на форму авторизации на рис. 2.28

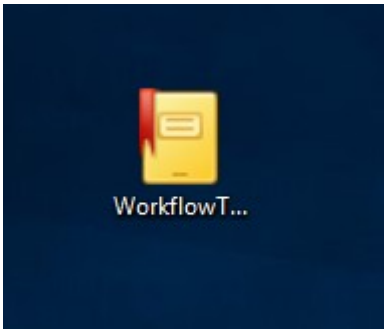


Рисунок 2.28 – Запуск программы

Далее у нас появляется форма авторизации пользователей, согласно определенным ранее ролям прав доступа в пункте 2.1.3. Здесь следует ввести логин и пароль, используемый в системе. На рисунке 2.29 показано меню авторизации пользователей.

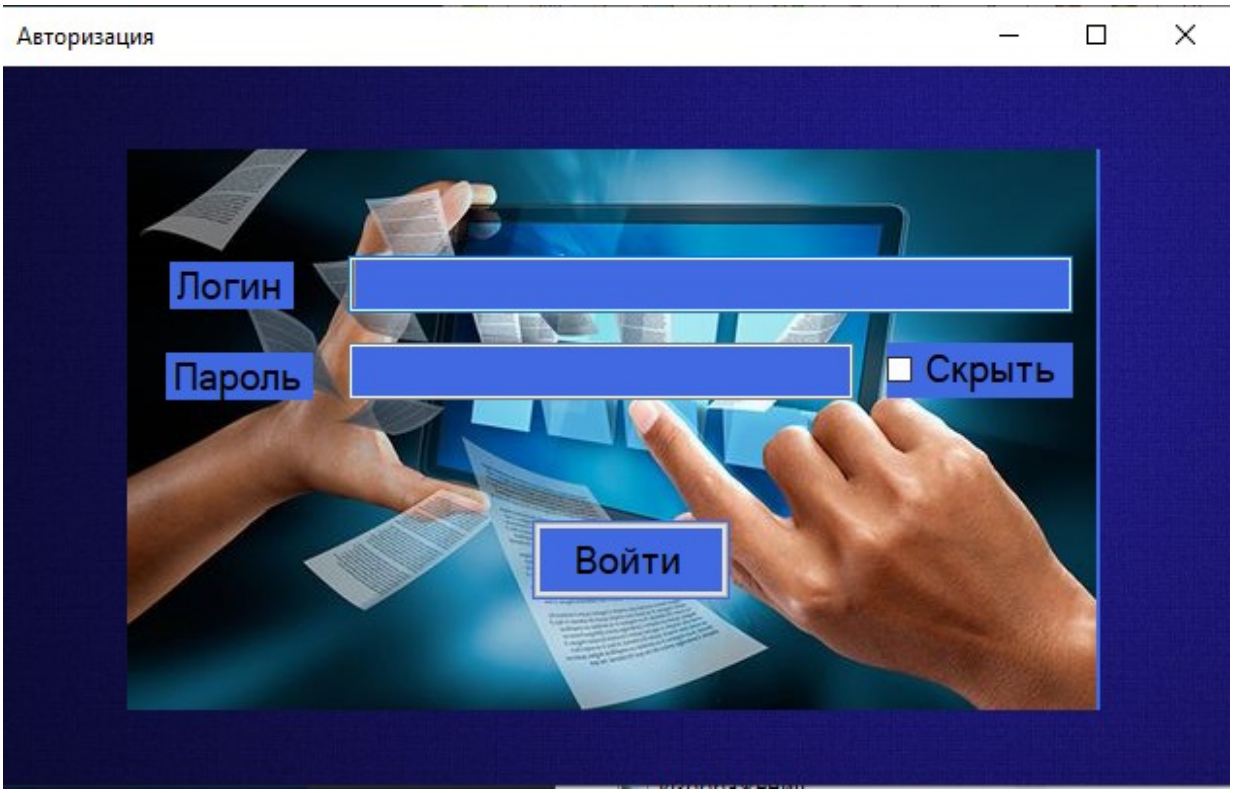


Рисунок 2.29 – Форма авторизации

После входа в систему мы попадем на главную форму на рис. 2.30

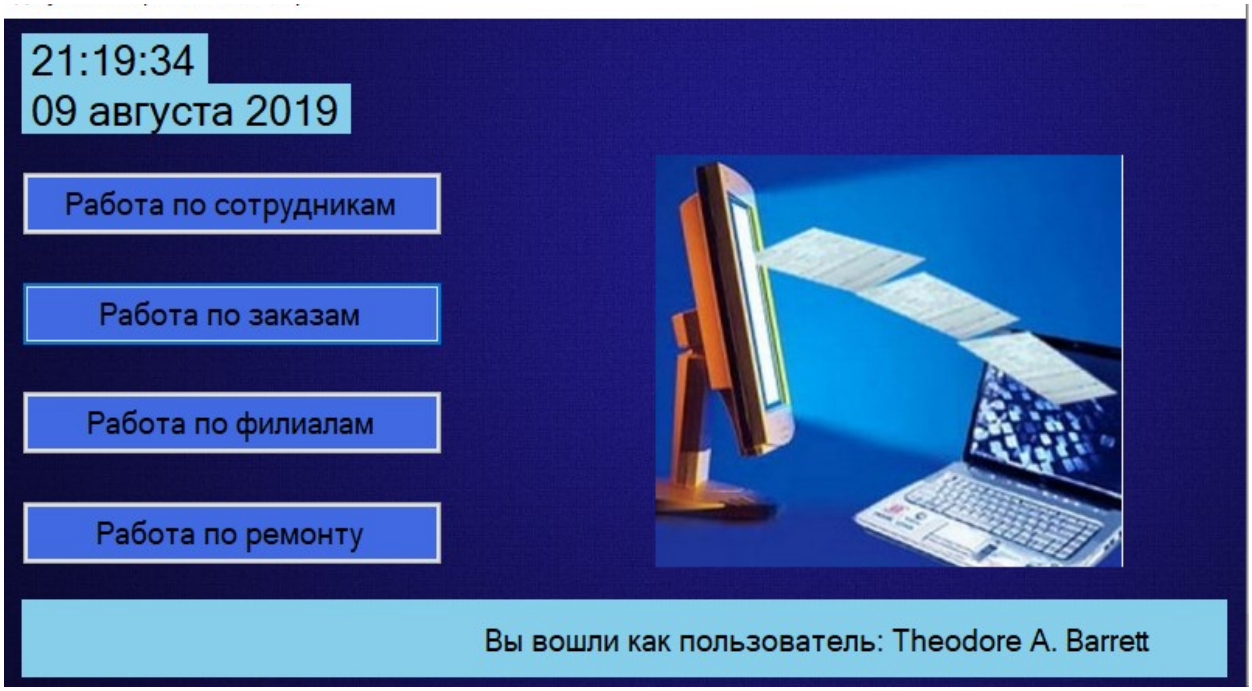


Рисунок 2.30 – Главная форма

При нажатии на кнопку, работа по сотрудникам, откроется форма сотрудники, на рис. 2.31 на которой будет показано топ 5 сотрудников по заказам в этом месяце.

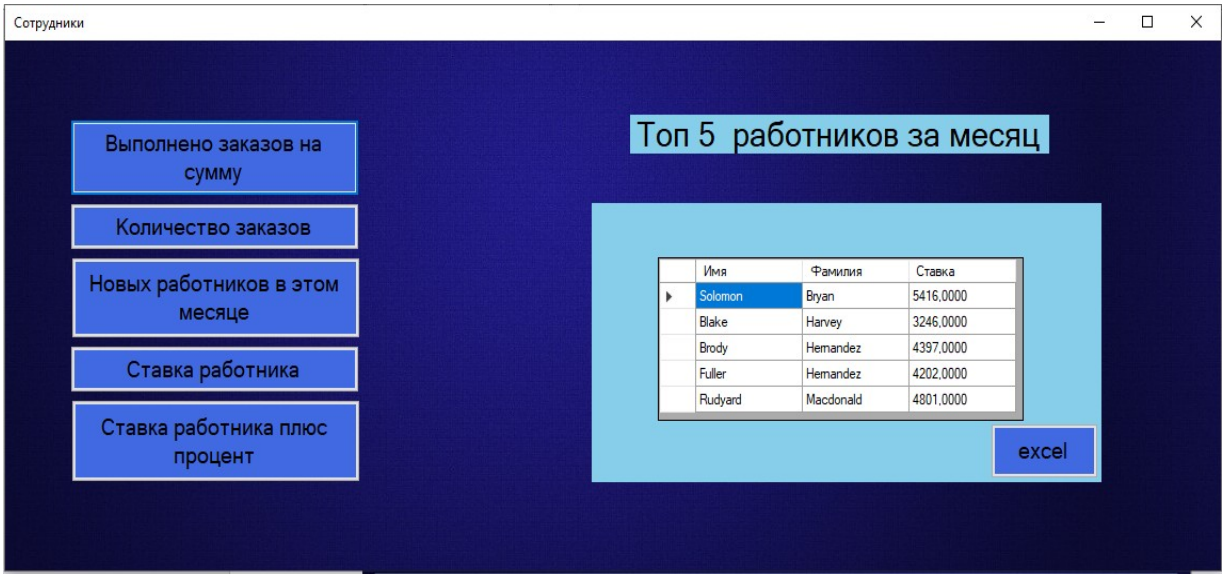


Рисунок 2.31 – Работа по сотрудникам

При нажатии на кнопку количество заказов будет выведена информация по сотрудникам на рис. 2.32

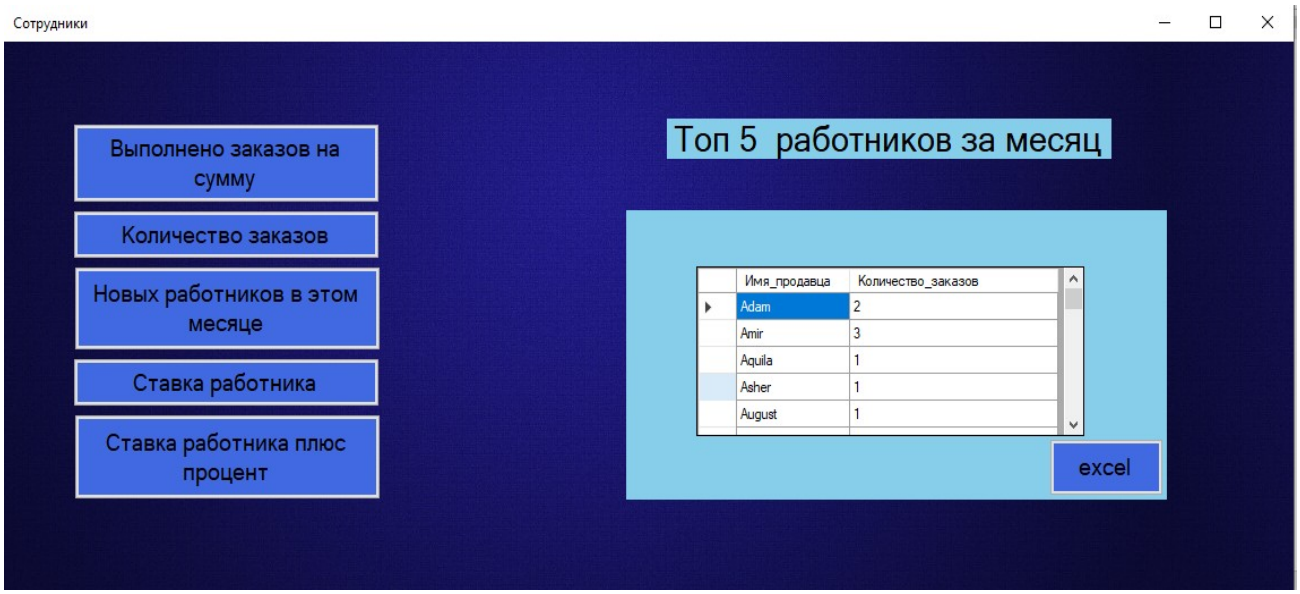


Рисунок 2.32 – Количество заказов

При нажатии на кнопку excel будет открыт в Excel на рис. 2.33

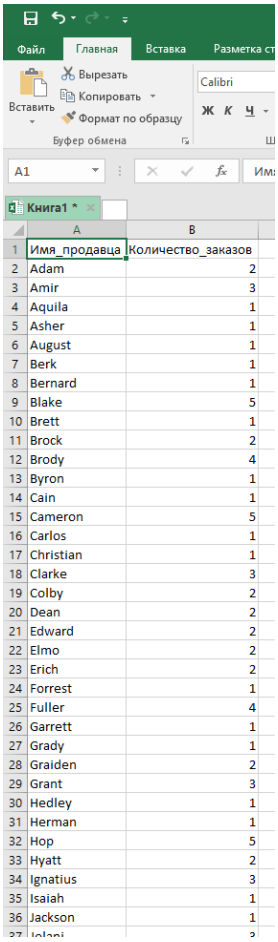


Рисунок 2.33 – Excel-файл

При нажатии на кнопку. работа по заказам, будет выведено окно с учетом налогов, зарплаты и доходов на рис. 2.34

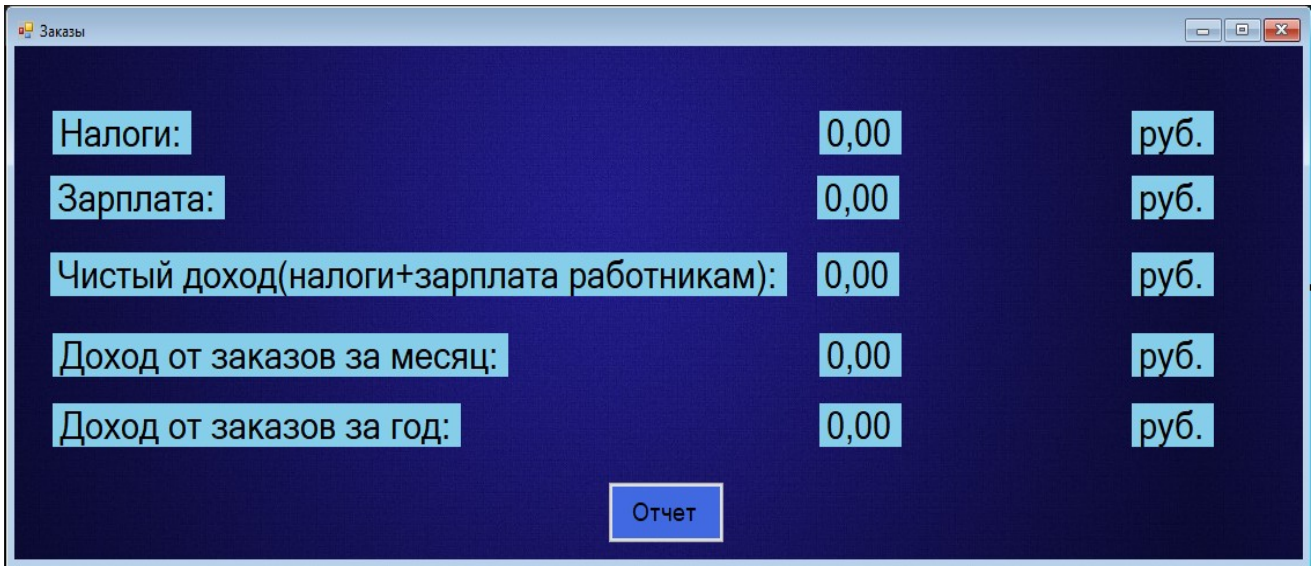


Рисунок 2.34 – Количество заказов

При нажатии на кнопку, работа по филиалам, будет выведено окно с информацией по филиалам на рис. 2.35

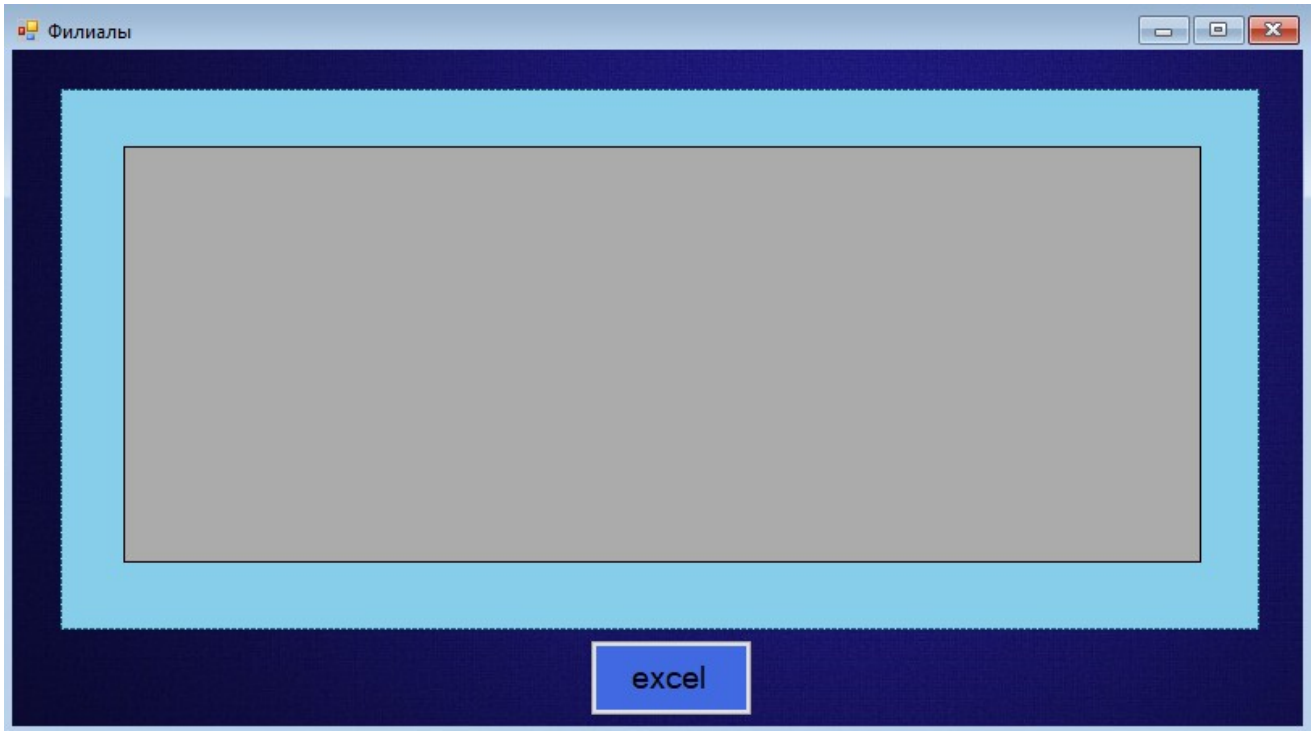


Рисунок 2.35 – Работа по филиалам

Листинг программного кода приведен в Приложении 1.

III Обоснование экономической эффективности проекта

3.1 Выбор и обоснование методики расчёта экономической эффективности

Стоимость информационных продуктов или услуг - это выраженная в денежной форме совокупность затрат на все работы, которые выполняются информационным органом в процессе сбора, переработки, хранения и реализации информации.

Стоимость программных средств делится на плановую и реальную.

Реальная стоимость программного продукта заключается в определении уже затраченных ресурсов на разработку приложения.

Самый простой способ определения себестоимости продукции - калькуляция. Для расчета себестоимости используются два показателя:

– показатель себестоимости каждого вида информационной продукции в соответствии с калькуляционным статьям расходами;

– сметная стоимость всей информационной продукции информационного органа по статьям расходов.

Для расчета стоимости программного продукта необходимо, прежде всего, установить трудоемкость его разработки. Для этого используют типовой состав этапов и увеличены нормативы времени на разработку программных средств. Согласно каталогов определяется объем программных средств в тысячах условных машинных команд программы-аналога [19], как показано в таблице 3.1.

Таблица № 3.1

Программные средства

Перечень программных средств	**Объем функции программных средств, условных машинных команд**
1 Техническое задание	
1.1 Структура входных и выходных данных	700

Продолжение таблицы № 3.1

Перечень программных средств	**Объем функции программных средств, условных машинных команд**
Всего: объем функции технического задания	700
2 Технический проект:	
2.1 Выбор языка программирования	700
2.2 Алгоритм, блок-схема	500
2.3 Программное средство ввода информации	300
Всего: объем функции технического проекта	1500
3 Рабочий проект:	
3.1 Корректировка	700
3.2 Оптимизация	500
3.3 Отладка	500
Всего: объем функции рабочего проекта	1700
4 Воплощение:	
4.1 Инструктаж	200
4.2 Сопровождение	100
Всего: объем функции воплощения	300
всего:	2700

Трудоемкость разработки программного продукта определяется на каждом этапе отдельно на основе определенной трудоемкости с учетом сложности разработки, степени новизны и использование стандартных модулей.

Трудоемкость технического задания определяется по формуле (3.1):

$$T_{ТЗ} = T^{K}_{P_{ТЗ}} ¿ L_1 \cdot K_н, \quad (3.1)$$

где $T^{K}_{P_{ТЗ}}$ - укрупненная норма времени на разработку ПО, которая корректируется коэффициентом с учетом условий разработки ПО, то есть в условиях компьютера, чел / час;

L_1 - удельный вес разработки этапа технического задания;

$K_н$ - поправочный коэффициент, учитывающий степень новизны.

$$T_{ТЗ} = 145 * 0,15 * 1 = 22 \text{ ч}$$

Трудоемкость технического проекта определяется по формуле (3.2):

$$ТТП = ТКРрп * L2 * Кн, \quad (3.2)$$

где L2 - удельный вес разработки этапа технического проекта.

$$ТТП = 442 * 0,16 * 1 = 70 \text{ ч}$$

Трудоемкость рабочего проекта определяется по формуле (3.3):

$$Трп = tкв * L3 * Кн * kт, \quad (3.3)$$

где L3- удельный вес разработки этапа рабочего проекта;

Кн - поправочный коэффициент, учитывающий степень использования в разработке типовых программ.

$$Трп = 483 * 0,36 * 1 * 0,5 = 86 \text{ ч}$$

Трудоемкость настройки и внедрения программного продукта определяется по формуле (3.4):

$$ТВ = ТkВ * L4 * Кн, \quad (3.4)$$

где L4- удельный вес этапа воплощения программного продукта.

$$Тв = 170 * 0,09 * 1 = 15 \text{ ч}$$

Общая трудоемкость разработки программного продукта определяется по таблице 3.2

Таблица № 3.2

Общая трудоемкость разработки программного продукта

Этапы разработки программного продукта	**Норма времени, ч**
1 Техническое задание ТТЗ 2 Технический проект (алгоритм, блок-схема) ТТП 3 Рабочий проект (составление программы) Трп 4 Настройка и внедрение ТВ	22 70 86 15
5 Пояснительная записка Тпз	45
Всего в т.ч .: - на разработку - на контроль руководителя - нормоконтроль Вместе:	283 17 1 256

Расходы на создание этой база данных можно разделить на текущие и капитальные. К текущим расходам относятся расходы, произошедшие в течение года в результате ее эксплуатации. К капитальным расходам относятся средства в форме капитальных вложений на создание и функционирование программного продукта. [19]

"Материалы" содержит затраты на материалы. Расчет материальных затрат выполняется в таблице 3.3 на основании расходных норм и цен:

Таблица № 3.3

Расчет материальных затрат

Наименование материальных затрат	**Количество**	**Цена, руб.**	**Стоимость, руб.**
Диск CD-R	2	11	22
Электроэнергия	75	3,36	252
Бумага (А4)	500	0,47	235
Чернила черные	1	100	100
Чернила цветные	0,2	150	30
Файлы	15	1	15
Переплет	1	45	45
Папка	1	15	15
всего:			714
Транспортно-заготовительные расходы			50
Вместе:			764

«Основная заработная плата» содержит расходы на оплату труда исполнителей и определяется по формуле (3.5):

$$З_о = \sum T_{общ} * С_{мз} * Ккпсо, \qquad (3.5)$$

где $\sum T_{общ}$- общая трудоемкость программного продукта, мес;

Смз - ставка минимальной заработной платы, руб .;

Ккпсо - коэффициент повышения ставок и окладов.

Со = 2,1 * 3723 * 1 = 7818 руб.

Общая трудоемкость программного продукта по месяцам определяется по формуле (3.6):

$$ТППобщ = \frac{\sum T_{общ} * Кмис}{Тпд * Кпер * Дк}, \quad (3.6)$$

где Кмис = 12 мес - количество месяцев в плановом периоде;

tпд = 8 ч - продолжительность рабочего дня;

Кпер = 0,73 - коэффициент перевода в календарные дни;

Дк = 360 дней - количество календарных дней в плановом периоде.

$$ТППобщ = \frac{256 * 12}{8 * 0,73 * 360} = 2,1 \ мес$$

Дополнительная заработная плата (оплата отпусков, времени выполнения государственных обязанностей, доплаты за выполнение дополнительных функций) исчисляется в процентах от основной заработной платы по формуле (3.7):

$$Здод = 0,1 * Со, \quad (3.7)$$

где 0,1 = 10% - коэффициент дополнительной заработной платы.

Здод = 0,1 * 7818 = 781,8 руб.

«Отчисления на социальные нужды», включая отчисления в пенсионный фонд в размере 22% от основной и дополнительной Зп, вычисляется по формуле 3.8.

$$Всоц = 22\% (Со + Здод), \quad (3.8)$$

Всоц =0,22 * (7818 + 781,8) = 1715,8 руб.

"Стоимость машинного времени" рассчитывается исходя из цены машино-часа и трудоемкости настройка и внедрение программы по формуле (3.9):

$$Змч = Вмч * ТВ, \quad (3.9)$$

где Вмч - стоимость машино-времени, руб .;

ТВ - трудоемкость отладки и внедрения программы, ч.

ОМЧ = 3 * 15 = 45 руб.

"Общехозяйственные расходы" - это расходы на зарплату с отчислениями управленческого персонала, специалистов, обслуживающего персонала, расходы, амортизацию зданий и др., определяются по формуле (3.10):

$$Взв = 0,3 * (Со + Здод), \quad (3.10)$$

где 0,3 - коэффициент пропорционального распределения общехозяйственных расходов.

$$Взв = 0,3 * (7818 + 781,8) = 2814,5 \text{ руб.}$$

Производственная себестоимость программного продукта определяется суммой рассчитанных статей (формула 3.11):

$$Ствир = МВ + Со + Здод + Всоц + ОМЧ + взв, \quad (3.11)$$

$$Ствир = 764 + 7818 + 781,8 + 1715,8 + 45 + 2815,5 = 13940,1 \text{ руб.}$$

Статья "Коммерческие расходы" связаны с расходами на изучение рынка, на рекламу и продажу продукции, они распределяются пропорционально производственной себестоимости по формуле (3.12):

$$Вк = 0,05 * Ствир, \quad (3.12)$$

где 0,05- коэффициент пропорционального распределения расходов.

$$Вк = 0,05 * 13940,1 = 697,005 \text{ руб.}$$

Плановая полная себестоимость программного продукта определяется по формуле (3.13):

$$Спов = Ствир + Вк, \quad (3.13)$$

$$Исп = 13940,1 + 697,005 = 14637,105 \text{ руб.}$$

Выполнены расчеты статей калькуляции приводятся в таблице 3.4

Таблица № 3.4

Калькуляция программного продукта

Статьи калькуляции	**Расходы, руб.**
1. Материалы и энергия технологическая	764
2. Основная заработная плата	7818
3. Дополнительная заработная плата	781,8
4. Отчисления на социальное страхование	1715,8
5. Стоимость машинного времени	45
6. Общехозяйственные расходы	2814,5
Производственная себестоимость	13940,1
7. Коммерческие расходы	697,005
Плановая полная себестоимость	14637,105

Таким образом, полная себестоимость разработки «Внедрение автоматизированных систем управления организацией» на примере ООО «Техторг» с учетом общей трудоемкости, составила 14637,105.

После расчета и получения всех показателей, приведем наглядный пример снижения трудовых затрат, составив графическую диаграмму, она изображена на рисунке 3.1

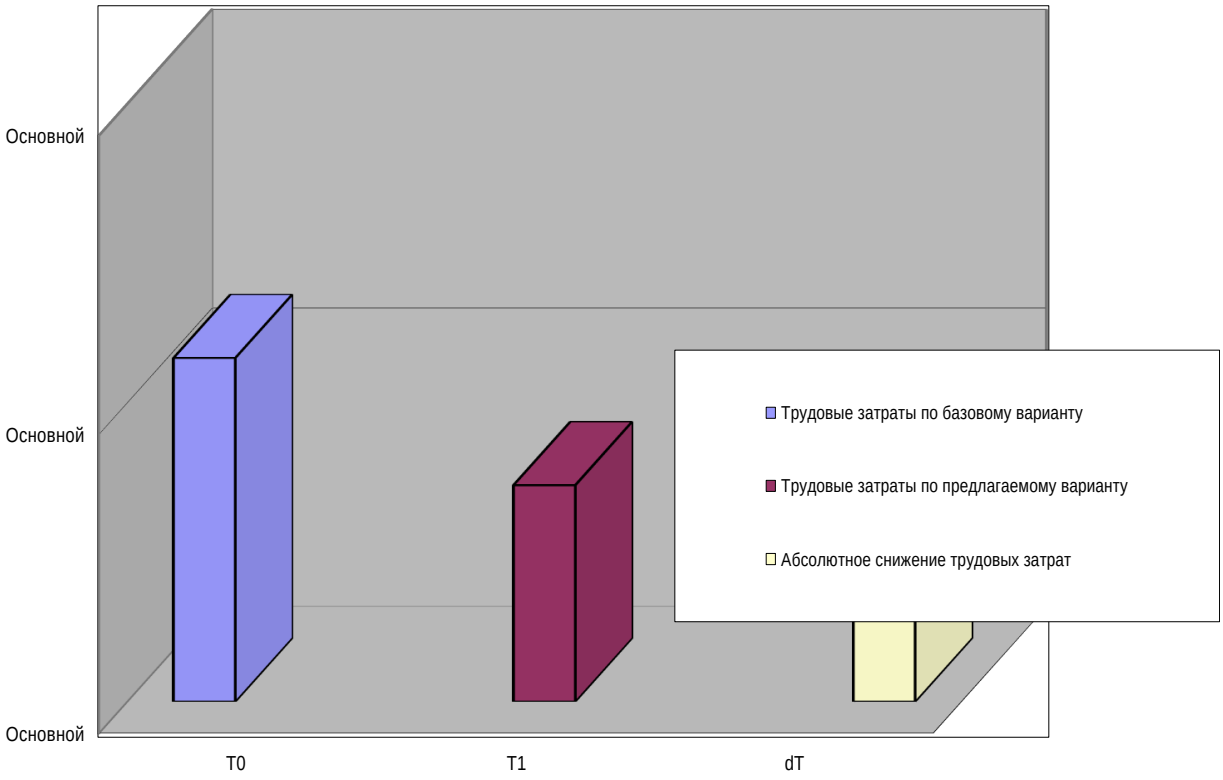


Рисунок 3.1 – Диаграмма снижения трудовых затрат

3.2 Расчёт показателей экономической эффективности проекта

Размер показателей включается в состав стоимости программной прибыли, определяется по формуле (3.14):

$$П = исп * ρ / 100\%, \quad (3.14)$$

где ρ = 20% - плановый процент прибыли.

$$П = 14637{,}105 * 20\% / 100\% = 2\ 927{,}421 \text{ руб.}$$

Цена реализации программного продукта состоит из полной себестоимости, прибыли и налогового долга (налога на добавленную стоимость) и определяется по формуле (3.15):

$$ЦР = исп + П + НДС, \quad (3.15)$$

где НДС = 20% - налог на добавленную стоимость

$$НДС = 0{,}2 * (Сэт + П)$$

$$НДС = 0{,}2 * (14637{,}105 + 2\ 927{,}421) = 3\ 512{,}90 \text{ руб.}$$

ЦР = 14637,105 + 2927,421 + 3512,90 = 21 077,43 руб.

Расчеты показали, что плановая цена на разработку «Внедрение автоматизированных систем документооборота предприятия ООО «Техторг» составляет 21 077,43 руб., которая значительно меньше, чем у конкурентов на рынке научно-технических разработок и информации, и свидетельствует о целесообразности разработки и продажи программного продукта, диаграмма снижения стоимостных затрат изображена на рисунке 3.2

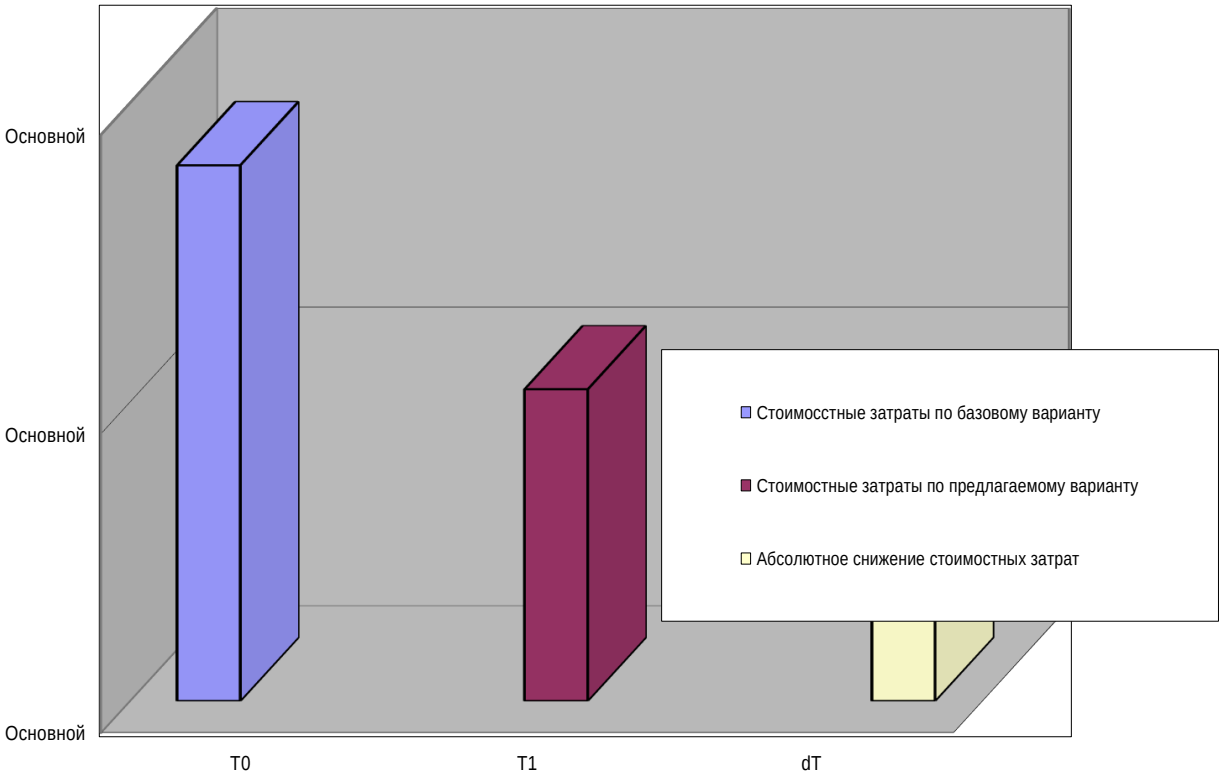


Рисунок 3.2 – Диаграмма снижения стоимостных затрат

Заключение

В наше время очень сложно, даже практически невозможно представить свою жизнь без различных современных технологий. Они стали неотъемлемой частью жизни людей и применяются в различной деятельности человека. Технический прогресс продолжает развиваться, и с каждым днём мы можем наблюдать за новинками и совершенствованиями в электронной технике, новыми открытиями в информационной сфере, большим влиянием информационных и интернет-технологий в жизни людей. Порой даже немыслимо осознавать то, что в прошлом человек жил без электронных технологий и доступа к ним. Если обратиться к научной точке зрения, то информационные технологии представляют весь накопленный опыт человечества в универсальном виде, пригодном для практического использования. Любая сфера деятельности в наши дни требует не только тщательного анализа эффективности труда, но и применение современных технологий управления различными ресурсами, благодаря которым стало возможным обеспечить наилучшее качество выполнения различных задач.

Еще одну сложность управления создает необходимость осуществления контроля выполнения различных работ, мониторинг ресурсов и прочего. Все эти процессы возможно упростить внедрив использование общего пространства обмена информацией, результатами выполнения задач и единой оценкой трудовых ресурсов. Благодаря всем этим мерам возможно осуществление рационального распределения рабочей силы с целью максимальной выгоды организации их исполнения.

Данный подход, который был осуществлен в процессе написания дипломного проекта обеспечит выполнение этих условий с применением самых современных технологий, таким образом управление организацией значительно облегчится, модернизируется и станет более продуктивным, что в свою очередь повысит эффективность всей компании в целом.

В проекте рассматриваются все этапы внедрения и доработки ИС, от аналитической разработки проекта, до внедрения проекта на предприятии.

Исходя из цели и поставленных задач, была определена структура данного дипломного проекта. Был произведён анализ деятельности компании и описание

услуг, анализ алгоритмов для создания системы, представлена программная реализация для написания приложения и описан результат работы системы.

Результатом решения поставленных задач является создание частичной автоматизированной обработки документов информационной системы.

В начале выполнения выпускной работы были изложены теоретические основы построения. Был выполнен комплекс мер, направленных на обоснование необходимости автоматизации: определена сущность задачи, описаны основные свойства системы, дано описание всех существующих бизнес-процессов, рассмотрены вопросы, связанные с анализом существующих разработок в этой области. Также в первой главе обосновываются проектные решения по информационному, программному, техническому и технологическому обеспечению.

Проектная часть посвящена рассмотрению этапов жизненного цикла проекта. Также дана характеристика информационной архитектуры разрабатываемого проекта, построена информационная модель задачи. Были рассмотрены различные методы приобретения программного продукта и выявлено наиболее подходящее решение. Рассмотрены наиболее важные этапы разработки проекта, определены технические и программные средства, необходимые для реализации данного проекта.

В ходе проектирования базы данных были пройдены все этапы проектирования: описание информационных объектов предметной области, проектирование инфологической модели предметной области, логическое и физическое проектирование БД. В качестве СУБД была выбрана MS Server.

Для разработки системы была выбрана концепция Windows-приложения с использованием Microsoft Entity Framework. В качестве среды разработки была выбрана среда Visual Studio.

В результате была создана программа, представляющая клиентское приложение для оформления и просмотра отчетности по документообороту:

- введена авторизация;
- разработана возможность просматривать работу по сотрудникам;
- разработана возможность отслеживания ремонтов.

Программа отвечает основным заявленным требованиям и реализует интуитивно понятный интерфейс.

Результатом проекта являются улучшения показателей работы компании, за счет автоматизации документооборота и управления взаимоотношениями с работниками компании ООО «Техторг».

В конце проекта приводятся данные об экономической эффективности проекта, на основании этих расчетов можно сделать вывод, что проект автоматизации задачи учета товаров и услуг в компании ООО «Техторг» является быстро окупаемым и целесообразным.

Результатом проекта являются улучшения показателей работы компании, за счет автоматизации документооборота ООО «Техторг».

Список литературы

1. «Основные понятия методологии и языка IDEF0» ГОССТАНДАРТ РОССИИ Руководящий документ IDEF0-2000. Методология функционального моделирования IDEF0. Принят и введен в действие 2000 г.

2. Мишенин А. И. «Теория экономических информационных систем» / Москва: Финансы и статистика, 2007г. – 240 с.: ил.;

3. Микрюков В. Ю. «Информация, информатика, компьютер, информационные системы, сети» / Москва: Феникс, 2007г. – 448 с.: ил.;

4. Информационная система — Wikimedia Foundation https://dic.academic.ru/dic.nsf/ruwiki/101791#cite_note-6

5. Остроух А. В. «Ввод и обработка цифровой информации» / Академия - Москва, 2012. - 288 с.

6. Коробов Н. А. «Информационные технологии в торговле» / Н.А. Коробов, А.Ю. Комлев. - М.: Академия, 2016. - 176 с.

7. Криницкий, Н.А. «Автоматизированные информационные системы» / Н.А. Криницкий, Г.А. Миронов, Г.Д. Фролов. - М.: Наука, 2017. - 382 с.

8. НОУ «Интуит» Лекция: Классификация программного обеспечения https://www.intuit.ru/studies/courses/3632/874/lecture/14291

9. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология (ИТ). Системная и программная инженерия. Процессы жизненного цикла программных средств. Руководящий документ Национальный стандарт РФ. Дата введения 2012-03-01

10. Майкл С. В. Тернер "Основы Microsoft Solution Framework" / Майкл С.В. Тернер. - М.: Русская Редакция, Питер, 2008. - 336 с.

11. Ицхак Калдерон Адизес «Управление жизненным циклом корпораций» / Манн, Иванов и Фербер, 2014. – 700 с. : ил.;

12. Избачков С.Ю., Петров В.Н. «Информационные системы» / СПб.: Питер, 2008. – 655 с

13. Гулиян Г.Б., Нестеров И.А. «Основы организации компьютерных сетей. Часть 1.» / Московская финансово-промышленная академия. – М., 2007г. – 169 с. ил.;

14. Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. «Базы данных» / Учебник, Корона – Век, 2010г. – 736 с.: ил.;

15. Томас Коннолли, Каролин Бегг «Базы данных. Проектирование, реализация и сопровождение. Теория и практика» / Вильямс 2016 г. -240 с.: ил;

16. Крис Дейт «Введение в системы баз данных» / Диалектика 2016 г. – 20 с.: ил.;

17. Тарасов Сергей Витальевич «Сергей Тарасов: СУБД для программиста. Базы данных изнутри» / Солон-пресс 2015 г. – 60 с.: ил.;

18. Руководство по Entity Framework – Введение в Entity Framework https://metanit.com/sharp/entityframework/1.1.php

19. О. С. Сухарев «Теория эффективности экономики” / КУРС, Инфра-М 2015 г. - 40 с.: ил..

ПРИЛОЖЕНИЕ 1 Листинг программного кода

```
usingSystem;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;

namespaceWorkflowTehtorg
{
publicpartialclassForm1 : Form
   {
public Form1()
      {
InitializeComponent();

DBWorkflow DB = newDBWorkflow();




      }

privatevoid button1_Click(object sender, EventArgs e)
      {

DBWorkflow DB = newDBWorkflow();

var login = DB.Administrators.ToList();
foreach (var email in login)
         {
if (email.email==textBox1.Text&&email.Password==textBox2.Text)
               {
GlobalVars.KeyAvtor = 1;
GlobalVars.AdminID = email.AdministratorId;
               }

               }


if (GlobalVars.KeyAvtor == 1)
         {
            Form2 f2 = newForm2();
f2.Show();
//MessageBox.Show("Admin ID = " + GlobalVars.AdminID);

}
elseMessageBox.Show("Нет такого пользователя");

}
```

```
privatevoid checkBox1_CheckedChanged(object sender, EventArgs e)
      {

if (checkBox1.Checked == true)
         {
            textBox2.UseSystemPasswordChar = true;

         }
else
         {
            textBox2.UseSystemPasswordChar = false;
         }
      }
   }
}


using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;

namespaceWorkflowTehtorg
{
publicpartialclassForm2 : Form
   {
public Form2()
      {
InitializeComponent();
DBWorkflow DB = newDBWorkflow();
varselectAdmin = from user inDB.Administrators
whereuser.AdministratorId == GlobalVars.AdminID
selectuser.Name;
         label3.Text = "Вывошликакпользователь: " + selectAdmin.First();

}

privatevoid timer1_Tick(object sender, EventArgs e)
      {
         label1.Text = DateTime.Now.ToString("T");
         label2.Text = DateTime.Now.ToString("dd MMMM yyyy");

if(label3.Left > -label3.Width)
         {
            label3.Left -= 15;
         }
```

```
else
        {
           label3.Left = panel1.Width;
        }

     }

privatevoid Form2_Load(object sender, EventArgs e)
     {
timer1.Start();
     }

privatevoid button1_Click(object sender, EventArgs e)
     {
        Form3 f3 = newForm3();
f3.Show();
     }

privatevoid button2_Click(object sender, EventArgs e)
     {
        Form4 f4 = newForm4();
f4.Show();
     }

privatevoid button3_Click(object sender, EventArgs e)
     {
        Form5 f5 = newForm5();
f5.Show();
     }

privatevoid button4_Click(object sender, EventArgs e)
     {
        Form6 f6 = newForm6();
f6.Show();
     }
  }
}


using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;
usingMicrosoft.Office.Interop.Excel;


namespaceWorkflowTehtorg
{
publicpartialclassForm3 : Form
  {
```

```
DBWorkflow DB = newDBWorkflow();
public Form3()
      {
InitializeComponent();
//Топсотрудники
varselectSellerOrd = from s inDB.Sellers
join o inDB.Ordersons.SellerIdequalso.SellerId
whereo.DateOrders.Value.Year == DateTime.Today.Year&&o.DateOrders.Value.Month == DateTime.-
Today.Month

selectnew {
Код_продавца =s.SellerId,
Имя = s.Name,
Фамилия = s.Surname,
Ставка =s.Salary,
Дата_заказа = o.DateOrders,
Сумма_заказа = o.OrderPrice,
Код_заказа =o.OrderId };

varplayersPerTeam = from admin inselectSellerOrd

group admin byadmin.Код_продавцаintoadminGroup

selectnew
                        {
Код_продовца = adminGroup.Key,
Количество_заказов = adminGroup.Count(),
                        };


/*//vardist=playersPerTeam.Distinct();*/

var sort = fromsortyinplayersPerTeam
//join o in selectSellerOrd on sorty.Код_продовца equals o.Код_продавца
orderbysorty.Количество_заказовdescending
selectnew {
Код_продавца = sorty.Код_продовца,
Количество_заказов = sorty.Количество_заказов,

};
var p = sort.ToList();
int p1 = p[0].Код_продавца;
intp2 = p[2].Код_продавца;
intp3 = p[3].Код_продавца;
intp4 = p[4].Код_продавца;
intp5 = p[5].Код_продавца;

varselectSeller = from s inDB.Sellers
join o inDB.Ordersons.SellerIdequalso.SellerId
wheres.SellerId==p1 || s.SellerId==p2 || s.SellerId == p3 || s.SellerId == p4 || s.SellerId == p5
selectnew
                         {

Имя = s.Name,
Фамилия = s.Surname,
```

```
Ставка = s.Salary,

                        };

vardisti = selectSeller.Distinct();

var pr1Q = disti.Take(300).ToList();

        dataGridView1.DataSource = pr1Q;



    }

privatevoid button1_Click(object sender, EventArgs e)//Суммазаказовзамесяц
    {
varslectSumSelers = (
from s inDB.Sellers
join o inDB.Ordersons.SellerIdequalso.SellerId
whereo.DateOrders.Value.Year == DateTime.Today.Year&&o.DateOrders.Value.Month == DateTime.-
Today.Month
groupsbynew
            {

s.SellerId,
s.Name,
s.Surname,
o.DateOrders,
o.OrderPrice

            } intoselerGroup
selectnew
            {
Код_продавца = selerGroup.Key.SellerId,
Имя = selerGroup.Key.Name,
Фамилия = selerGroup.Key.Surname,
Дата_заказа = selerGroup.Key.DateOrders,
Сумма = selerGroup.Key.OrderPrice
            }

            );


var pr1Q1 = slectSumSelers.Take(300).ToList();
        dataGridView1.DataSource = pr1Q1;
    }

privatevoid button6_Click(object sender, EventArgs e)//excel
    {
try
        {
Microsoft.Office.Interop.Excel.Application excel = newMicrosoft.Office.Interop.Excel.Application();
excel.Visible = true;
Microsoft.Office.Interop.Excel.Workbook workbook = excel.Workbooks.Add(System.Reflection.Miss-
ing.Value);
```

```
Microsoft.Office.Interop.Excel.Worksheet sheet1 = (Microsoft.Office.Interop.Excel.Worksheet)work-
book.Sheets[1];
intStartCol = 1;
intStartRow = 1;
int j = 0, i = 0;

//Write Headers
for (j = 0; j <dataGridView1.Columns.Count; j++)
          {
Microsoft.Office.Interop.Excel.RangemyRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow, StartCol + j];
             myRange.Value2 = dataGridView1.Columns[j].HeaderText;
          }

StartRow++;

//Write datagridview content
for (i = 0; i<dataGridView1.Rows.Count; i++)
          {
for (j = 0; j <dataGridView1.Columns.Count; j++)
             {
try
                {
Microsoft.Office.Interop.Excel.RangemyRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow + i, StartCol + j];
                   myRange.Value2 = dataGridView1[j, i].Value == null ? "" : dataGridView1[j, i].Value;
                }
catch
                {
                   ;
                }
             }
          }
       }
catch (Exception ex)
       {
MessageBox.Show(ex.ToString());
       }
    }

privatevoid button2_Click(object sender, EventArgs e)//Количествозаказовзамесяц
    {
varselectSellerOrd = from s inDB.Sellers
join o inDB.Ordersons.SellerIdequalso.SellerId
whereo.DateOrders.Value.Year == DateTime.Today.Year&&o.DateOrders.Value.Month == DateTime.-
Today.Month

selectnew
                        {
Код_продавца = s.SellerId,
Имя = s.Name,
Фамилия = s.Surname,
Ставка = s.Salary,
Дата_заказа = o.DateOrders,
Код_заказа = o.OrderId
```

```
                    };

varplayersPerTeam = from admin inselectSellerOrd

group admin byadmin.Имяintoadmin Group
selectnew
                    {
Имя_продавца = adminGroup.Key,
Количество_заказов = adminGroup.Count(),
                    };
var pr1Q1 = playersPerTeam.Take(300).ToList();
        dataGridView1.DataSource = pr1Q1;

    }

privatevoid button5_Click(object sender, EventArgs e)//новыеработники в этоммесяце
    {
varselectSelers = from s inDB.Sellers
wheres.HiringDate.Value.Year == DateTime.Today.Year&&s.HiringDate.Value.Month == DateTime.-
Today.Month
selectnew
                    {
Имя = s.HiringDate,
Отчество = s.Patronymic,
Номер_телефона = s.Phone,
//Ставка = s.Salary
};
var pr1Q1 = selectSelers.Take(300).ToList();
        dataGridView1.DataSource = pr1Q1;
    }

privatevoid button4_Click(object sender, EventArgs e)//Ставкаработника
    {
varselectSelers = from s inDB.Sellers
selectnew
                    {
Имя = s.HiringDate,
Отчество = s.Patronymic,
Номер_телефона = s.Phone,
Ставка = s.Salary
};
var pr1Q1 = selectSelers.Take(300).ToList();
        dataGridView1.DataSource = pr1Q1;
    }

privatevoid button3_Click(object sender, EventArgs e)//ставка и процент
    {
varselectSelers = from s inDB.Sellers
selectnew
                    {
Имя = s.HiringDate,
Отчество = s.Patronymic,
Номер_телефона = s.Phone,
Ставка = s.Salary+(s.Salary*(decimal)0.1)
                    };
```

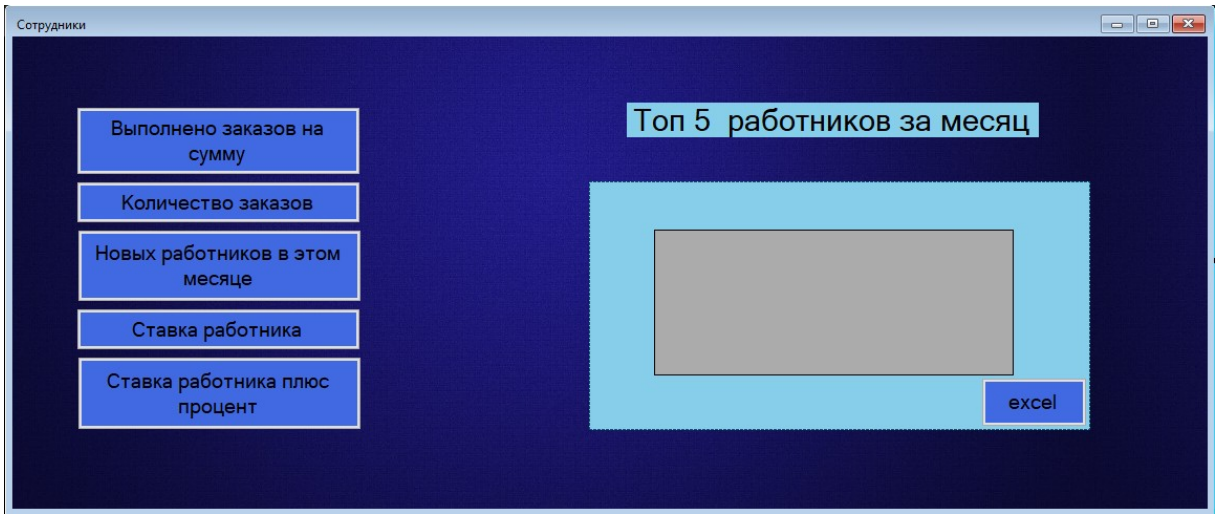
```
var pr1Q1 = selectSelers.Take(300).ToList();
            dataGridView1.DataSource = pr1Q1;
        }


    }
}


usingMicrosoft.Office.Interop.Word;
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;
using Word = Microsoft.Office.Interop.Word;
usingSystem.Reflection;

namespaceWorkflowTehtorg
{
publicpartialclassForm4 : Form
    {
DBWorkflow DB = newDBWorkflow();
public Form4()
        {
InitializeComponent();
        }

privatevoid Form4_Load(object sender, EventArgs e)
        {
varsumCash=  DB.Orders.Where(a=>a.DateOrders.Value.Year == DateTime.Today.Year&&a.Date-
Orders.Value.Month == DateTime.Today.Month).Sum(a =>a.OrderPrice );//замесяц
//MessageBox.Show(sumCash.ToString());
            label6.Text = (sumCash.Value * (decimal)0.2).ToString();

varsumSalary = DB.Sellers.Sum(a =>a.Salary);
            label7.Text = (sumSalary).ToString();

            label9.Text = (sumCash - (decimal)(float.Parse(label6.Text) +
float.Parse(label7.Text))).ToString();

            label4.Text = sumCash.ToString();

varsumCashYear = DB.Orders.Where(a =>a.DateOrders.Value.Year == DateTime.Today.Year ).Sum(a
=>a.OrderPrice);//Загод
            label5.Text = sumCashYear.ToString();
        }

privatevoid button6_Click(object sender, EventArgs e)
        {
try
```

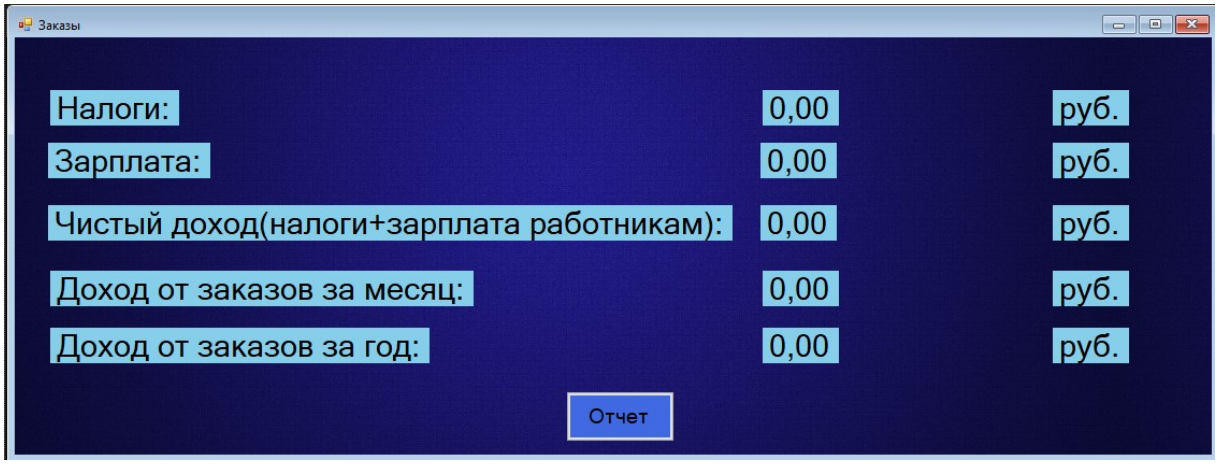
```
        {
Microsoft.Office.Interop.Excel.Application excel = newMicrosoft.Office.Interop.Excel.Application();
excel.Visible = true;
Microsoft.Office.Interop.Excel.Workbook workbook = excel.Workbooks.Add(System.Reflection.Miss-
ing.Value);
Microsoft.Office.Interop.Excel.Worksheet sheet1 = (Microsoft.Office.Interop.Excel.Worksheet)work-
book.Sheets[1];


//Write Headers

Microsoft.Office.Interop.Excel.RangemyRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[1,
1];
                myRange.Value2 = label1.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[1, 2];
                myRange.Value2 = label8.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[1, 3];
                myRange.Value2 = label10.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[1, 4];
                myRange.Value2 = label2.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[1, 5];
                myRange.Value2 = label3.Text;

//Write datagridview content
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[2, 1];
                myRange.Value2 = label6.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[2, 2];
                myRange.Value2 = label7.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[2, 3];
                myRange.Value2 = label9.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[2, 4];
                myRange.Value2 = label4.Text;
myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[2, 5];
                myRange.Value2 = label5.Text;


        }
catch (Exception ex)
        {
MessageBox.Show(ex.ToString());
        }

    }
  }
}
```

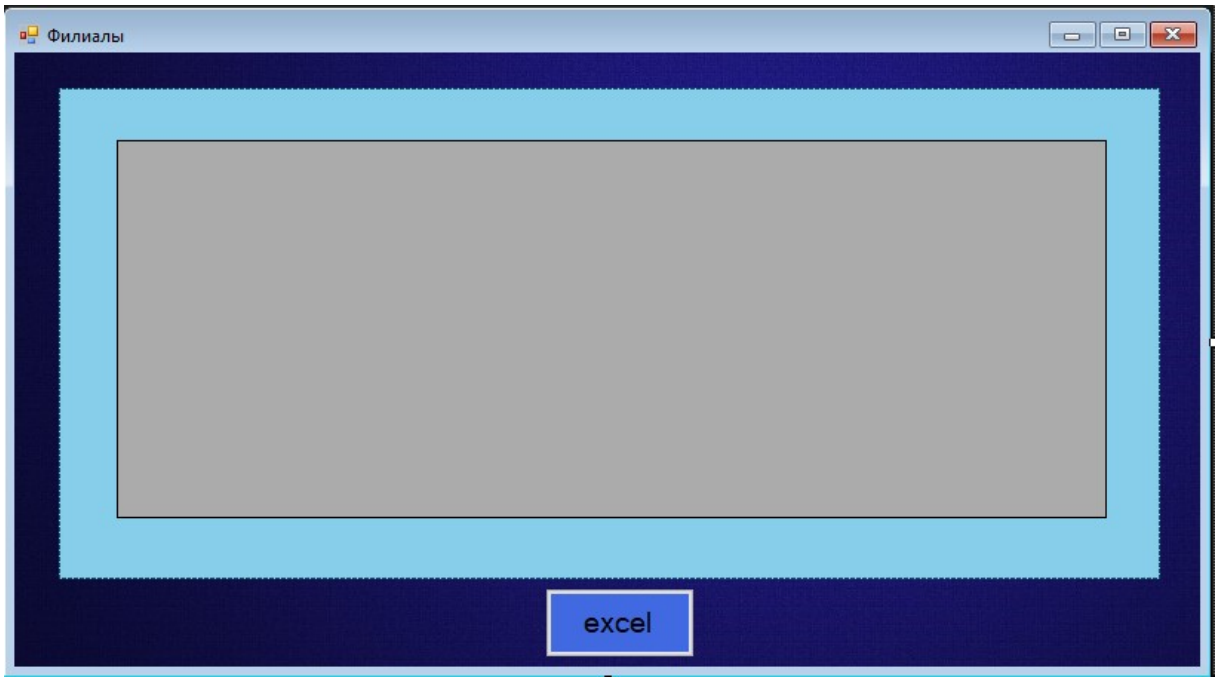
ПРИЛОЖЕНИЕ 2 Формы

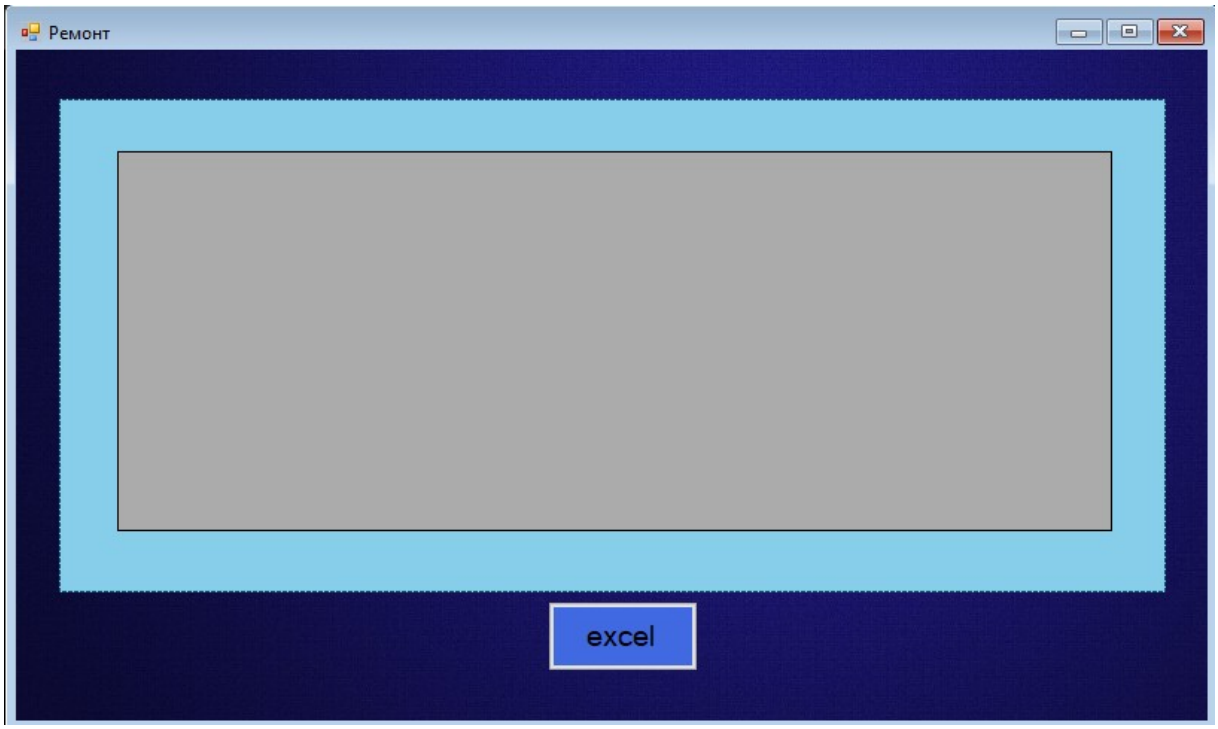





Сотрудники
Выполнено заказов на сумму
Количество заказов
Новых работников в этом месяце
Ставка работника
Ставка работника плюс процент
Топ 5 работников за месяц
excel


Заказы
Налоги:
0,00
руб.
Зарплата:
0,00
руб.
Чистый доход(налоги+зарплата работникам):
0,00
руб.
Доход от заказов за месяц:
0,00
руб.
Доход от заказов за год:
0,00
руб.
Отчет


Филиалы
excel


Ремонт
excel