

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Удмуртский государственный университет»
Филиал ФГБОУ ВО «УдГУ» в г. Воткинске
Кафедра информационных и инженерных технологий

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по подготовке и защите выпускных квалификационных работ
по направлению подготовки
09.03.03 «ПРИКЛАДНАЯ ИНФОРМАТИКА»
профиля подготовки «Прикладная информатика в экономике»



Ижевск
2023

УДК 004(075.8)
ББК 32.97р30
М545

Рекомендовано к изданию Учебно-методическим советом филиала ФГБОУ ВО «УдГУ» в г. Воткинске.

Рецензенты: канд. техн. наук, доцент каф. АМО ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова» А.Н. Домбрачев;
д-р техн. наук, профессор каф. ТБ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова» А.П. Тюрин.

Составители: Мамрыкин О.В., Кузнецов А.П., Раскин П.Н.

М545 Методические рекомендации по подготовке и защите выпускных квалификационных работ по направлению подготовки 09.03.03 «Прикладная информатика», профиля подготовки «Прикладная информатика в экономике»: [Электрон. ресурс] / сост. О.В. Мамрыкин, А.П. Кузнецов, П.Н. Раскин. – Ижевск : Удмуртский университет, 2023. – 95 с.

Объектом методических рекомендаций являются процессы подготовки и защиты выпускной квалификационной работы выпускника по направлению подготовки 09.03.03 «Прикладная информатика», профиль – «Прикладная информатика в экономике».

Цель работы – повышение эффективности подготовки выпускников. В процессе подготовки данных материалов использовались: «Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.03 «Прикладная информатика», «ГОСТ 7.32-2017 СИБИБД. Отчет о научно-исследовательской работе. Структура и правила оформления (с Поправками)», определены основные этапы подготовки и защиты выпускных квалификационных работ, даны углубленные инструкции выпускникам по прохождению данных этапов, разработаны образцы документов, требуемых выпускающей кафедрой.

УДК 004(075.8)
ББК 32.97р30

© О.В. Мамрыкин, А.П. Кузнецов,
П.Н. Раскин, 2023
© Филиал ФГБОУ ВО «УдГУ»
в г. Воткинске, 2023

ОГЛАВЛЕНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	4
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ.....	5
ВВЕДЕНИЕ.....	6
1 ПОДГОТОВКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ.....	8
1.1 Сбор материала для подготовки ВКР на объекте практики	8
1.2 Выбор и закрепление темы выпускной квалификационной работы	8
1.3 Написание и оформление выпускной квалификационной работы.	10
2 ОПИСАНИЕ ПРОЦЕССА ЗАЩИТЫ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ.....	12
2.1 Предварительная защита ВКР на выпускающей кафедре	12
2.2 Защита ВКР в Государственной экзаменационной комиссии.....	12
3 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ.....	14
4 ТРЕБОВАНИЯ К РАЗДЕЛАМ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ.....	16
4.1 Требования к содержанию раздела «Введение» выпускной квалификационной работы.....	16
4.2 Требования к содержанию раздела «Основная часть» выпускной квалификационной работы.....	16
4.3 Требования к содержанию раздела «Заключение» выпускной квалификационной работы.....	19
4.4 Требования к содержанию раздела «Список использованных источников» выпускной квалификационной работы	19
4.5 Требования к содержанию раздела «Приложения» выпускной квалификационной работы.....	19
4.6 Требования к объему выпускной квалификационной работы	19
5 РЕКОМЕНДАЦИИ ПО СТРУКТУРЕ КОМПЬЮТЕРНОЙ ПРЕЗЕНТАЦИИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ.....	20
ЗАКЛЮЧЕНИЕ	21
ПРИЛОЖЕНИЯ.....	22

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящей методической работе применяют следующие термины с соответствующими определениями.

Выпускная квалификационная работа – логически завершенное исследование по одной из актуальных тем в области прикладной информатики, в котором выпускник демонстрирует уровень овладения необходимыми теоретическими знаниями, практическими умениями и навыками, позволяющими ему самостоятельно решать профессиональные задачи

Тема выпускной квалификационной работы – выбирается обучающимся самостоятельно или по рекомендации научного руководителя. Тема должна подбираться в соответствии с необходимостью продемонстрировать в рамках выпускной квалификационной работы (ВКР) высокий итоговый квалификационный уровень обучающегося. Научная или прикладная актуальность темы **обязательны**. Тема согласовывается на заседании выпускающей кафедры и утверждается приказом директора ректора

Научный руководитель выпускной квалификационной работы – специалист в предметной области, по которой выполняется ВКР, обладающий опытом профессиональной деятельности в рамках выбранной тематики и обеспечивающий научное руководство подготовкой ВКР, оформлением ее результатов, а также осуществляющий консультационное обеспечение деятельности обучающегося в процессе его подготовки к защите ВКР. Научный руководитель утверждается на заседании выпускающей кафедры и утверждается приказом ректора

Выпускающая кафедра – основная структурная единица факультета в высшем учебном заведении, осуществляющая подготовку слушателей, студентов и курсантов в рамках определённой специализации

Предзащита ВКР – организационное мероприятие, позволяющая помочь студенту адаптироваться к процедуре защиты

Защита ВКР – итоговое публичное аттестационное мероприятие, позволяющее установить соответствие уровня подготовки выпускника требованиям ФГОС направления подготовки и квалификационным требованиям выбранного профиля подготовки. Проходит в форме научного доклада, защищающегося с последующим обсуждением под управлением Государственной экзаменационной комиссии

Государственная Экзаменационная комиссия (ГЭК) – комиссия в составе – председатель, члены комиссии и секретарь, определяет порядок прохождения защит ВКР и определяет итоговую оценку уровня квалификации выпускаемого специалиста. Состав комиссии и сроки ее работы утверждаются приказом ректора

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей методической работе применяют следующие сокращения и обозначения:

ВКР – выпускная квалификационная работа

ИС – информационные системы

СУБД – система управления базами данных

Филиал – филиал федерального государственного бюджетного образовательного учреждения высшего образования «Удмуртский государственный университет» в городе Воткинске

ФГОС – федеральный государственный образовательный стандарт

ГЭК – Государственная экзаменационная комиссия

ГИА – Государственная итоговая аттестация

ВВЕДЕНИЕ

Итоговая государственная аттестация обучающихся в филиале федерального государственного бюджетного образовательного учреждения высшего образования «Удмуртский государственный университет» в городе Воткинске (ФГБОУ ВО «УдГУ» в г. Воткинске) (далее – Филиал) по направлению подготовки 09.03.03 «Прикладная информатика» профиль подготовки «Прикладная информатика в экономике» осуществляется в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.03 «Прикладная информатика» (утв. приказом Министерства образования и науки РФ от 19 сентября 2017 г. № 922).

Итоговая государственная аттестация включает в себя защиту выпускной квалификационной работы.

Итоговую государственную аттестацию осуществляют Государственные экзаменационные комиссии (ГЭК), организуемые в вузе по соответствующему направлению на каждый календарный год.

Подготовка и защита выпускной квалификационной работы является заключительным этапом обучения обучающихся в Филиале и обязательной частью основной образовательной программы бакалавриата.

Главной целью итоговой государственной аттестации является установление соответствия уровня подготовки выпускников, сформированных у них общекультурных, общепрофессиональных и профессиональных компетенций требованиям ФГОС ВО.

Основными задачами подготовки выпускной квалификационной работы обучающегося являются систематизация, углубление и закрепление теоретических знаний и полученных во время обучения практических навыков в рамках самостоятельного решения рассматриваемой в ВКР конкретной проблемы в соответствии с видами послевузовской профессиональной деятельности. Независимо от темы ВКР, при ее выполнении и представлении на заседании ГЭК выпускник должен показать способность и умение профессионально излагать специальную информацию, презентовать полученные результаты, научно аргументировать и защищать свою точку зрения.

По результатам выполнения и защиты выпускной квалификационной работы обучающегося, ГЭК оценивает готовность обучающегося к самостоятельной профессиональной деятельности и присваивает ему квалификацию (степень) «бакалавр» по соответствующему направлению.

Выпускная квалификационная работа – это комплексная проектная работа, содержащая постановку и решение поставленной задачи, оформленное в виде технологических, программных и других проектных документов.

Главной целью и содержанием ВКР является системные новые решения в следующих областях профессиональной деятельности:

1. Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации информационных систем.

2. Проектирование информационных систем, программирование приложений, создание прототипов информационных систем и их модулей.
3. Управление жизненным циклом информационных систем и информационных технологий.
4. Программная поддержка процессов принятия решений для управления предприятием.
5. Сквозные технологии в промышленности (в сфере информатики и вычислительной техники).
6. Задачи, связанные с разработкой программных решений на основе машинного обучения, искусственного интеллекта, нейросетевых алгоритмов.
7. Интернет вещей

Выпускная квалификационная работа состоит из теоретических части, практических разработок и расчётов. В качестве тем, как правило, выбираются проблемы, с которыми специалист в области прикладной информатики будет встречаться в рамках практической деятельности, и которые соответствуют объёму и перечню теоретических знаний и практических навыков, полученных за время обучения в вузе.

Темами для ВКР могут быть задачи разработки, внедрения, реализация любых фаз жизненного цикла информационных систем:

1. Управление проектированием и создание новых информационных систем поддержки принятия управленческих решений.
2. Управление модернизацией и развитием существующих систем информатизации управленческой деятельности.
3. Управление проектами внедрения (развертывания и адаптации) готовых информационных систем.
4. Сопровождение и эксплуатация автоматизированных информационных систем.
5. Управление научными исследованиями, направленными на разработку и внедрение нового методического обеспечения и инструментальных средств в области информационных технологий.
6. Программная поддержка создания бизнес-ориентированных информационных систем.

Примерные темы выпускных квалификационных работ приведены в Приложении А.

При написании работы следует уделить особое внимание корректности заимствований и цитированию. Любое использование чужого текста, алгоритмов, таблиц, рисунков и диаграмм требует обязательной ссылки на первоисточник.

1. ПОДГОТОВКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Общее руководство и контроль за ходом написания ВКР осуществляет выпускающая кафедра. Решением кафедры каждому выпускнику назначается руководитель, что закрепляется приказом по высшему учебному заведению.

Студент может работать над своей будущей ВКР задолго до начала срока, определенного в календарном графике учебного плана, как время, необходимое для выполнения ВКР. Рекомендуется связать курсовой проект по дисциплине «Проектирование информационных систем» на 4 курсе с тематикой своей будущей ВКР и, таким образом, получить хороший задел, который в рамках ВКР будет развит. Также рекомендуется заранее выбрать себе руководителя ВКР и получить его согласие на руководство.

Следует подчеркнуть, что основной обязанностью руководителя является определение направления проектирования, выбор или согласование темы ВКР, консультирование, предостережение обучающегося от грубых ошибок. При этом руководитель не несет ответственности за ошибки в расчетах, недостатки в стиле и грамотности изложения материала, качества его оформления. Подпись руководителя удостоверяет, что работа выполнена в соответствии с требованиями ФГОС, самостоятельно и в соответствии с техническим заданием.

Подготовка ВКР включает в себя ряд этапов.

1.1 Сбор материала для подготовки ВКР на объекте практики

В течение преддипломной практики обучающийся обязан: ознакомиться с деятельностью предприятия (учреждения, фирмы), изучить основные направления его функционирования, подробно изучить реализуемый бизнес-процесс как объект управления и информатизации. Построить модели «AS-IS» основных и вспомогательных бизнес-процессов, используя актуальные методологии. Определить направления и вероятные темы ВКР. Собрать материалы, необходимые для написания ВКР:

- Описания и модели бизнес-процессов.
- Образцы или шаблоны документов, циркулирующих в рамках бизнес-процессов.
- Официальные и технические документы предприятия и другие.

1.2 Выбор и закрепление темы выпускной квалификационной работы

Тематика ВКР должна быть актуальна, соответствовать современному состоянию и перспективам развития ИС и ИКТ, а также методов анализа, проектирования таких систем. **При этом тематика обязана быть направлена на повышение эффективности бизнес-процессов предприятий, организации.**

В процессе работы над ВКР обучающийся показывает знания, умения и навыки по следующим направлениям:

1. Проведение обследования прикладной области в соответствии с профилем подготовки; моделирование прикладных и информационных процессов; формирование требований к прикладным информационным процессам; технико-экономическое обоснование проектных решений, составление технических заданий на автоматизацию и информатизацию решения прикладных задач, техническое проектирование ИС в соответствии со спецификой профиля подготовки; программирование, тестирование и документирование приложений; аттестация и верификация ИС;
2. Автоматизированное решение прикладных задач операционного и аналитического характера; информационное обеспечение прикладных процессов; внедрение, адаптация, настройка и интеграция проектных решений по созданию ИС; сопровождение и эксплуатации ИС;
3. Участие в организации и управлении информационными процессами, ресурсами, системами, сервисами; использование функциональных и технологических стандартов; обучение и консультирование пользователей в процессе эксплуатации ИС; участие в переговорах с заказчиком; презентация проектов;
4. Анализ прикладных процессов, разработка вариантов автоматизированного решения прикладных задач; анализ и выбор методов и средств автоматизации и информатизации прикладных процессов на основе современных информационно-коммуникационных технологий; оценка затрат и надежности проектных решений;
5. Применение системного подхода к автоматизации и информатизации решения прикладных задач, к построению информационных систем на основе современных информационно-коммуникационных технологий; подготовка обзоров, аннотаций, составление рефератов, документации в соответствии с действующими нормативными актами.

Тема предлагается обучающимся самостоятельно либо по рекомендации научного руководителя. Как правило, объектом выбранной темы является предприятие места прохождения преддипломной практики, а целью выполнения ВКР – одна из актуальных проблем этого предприятия, подпадающая под проблематику профессиональной деятельности выпускника (смотри «Введение» и приложение А). **Допускается обоснованный выбор темы, не привязанной к месту прохождения преддипломной практики.**

Закрепление за обучающимся темы производится по его личному заявлению на имя заведующего выпускающей кафедры (пример оформления приве-

ден в приложении Б). Заявление должно быть написано студентом и согласовано с руководителем **не менее чем за 3 месяца плюс 1 неделя до защиты ВКР.**

Заявления обучающихся с избранными темами выпускных квалификационных работ рассматриваются выпускающей кафедрой. Кафедра оформляет приказ о закреплении за обучающимися тем выпускных квалификационных работ, назначает руководителей. Приказ подписывается ректором (директором).

Обучающемуся, получившему приказ о закреплении за ним темы, и его руководителю рекомендуется составить и согласовать «Задание на выполнение ВКР». Примерная форма представлена в приложении Г. Задание на выполнение ВКР подписывает научный руководитель и обучающийся.

1.3 Написание и оформление выпускной квалификационной работы.

На выполнение ВКР выделяется 5 недель. За это время студент должен решить все задачи, предусмотренные темой ВКР, оформить пояснительную записку, подготовить доклад на защиту и презентацию к докладу.

В процессе работы над ВКР выпускающая кафедра составляет график смотров ВКР, с целью контроля хода выполнения работ над ВКР студентами, консультаций по возникающим вопросам. Явка студентов на смотры является обязательной. По результатам смотров заполняется ведомость явок с отметками о степени выполнения работ каждым студентом. Обычно проводится 3 смотра:

- первый смотр: студент показывает вводную часть своей ВКР и первую главу, где отражен обзор предметной области, обоснование выбора методов и средств достижения целей своей темы.
- второй смотр: студент показывает дополнительно 2 и 3 главы своей работы, а также практическую часть своей разработки (программный продукт, информационную систему, веб-приложение, сайт, базу данных, электронное устройство и его работу (если это связано с тематикой его работы).
- третий смотр является заключительным, студент приносит полностью готовую сброшюрованную пояснительную записку с подписей руководителя, отзывом руководителя, отчетом о проверке в системе Антиплагиат, а также предоставляет для демонстрации готовый программный продукт. На данном смотре производится экспертиза ВКР (нормоконтроль) и выносится решение о допуске студента до Государственной итоговой аттестации, о чем делается отметка на титульном листе пояснительной записки к ВКР.

Перед заключительным смотром законченная и подписанная обучающимся ВКР представляется руководителю. Если содержание пояснительной записки к ВКР соответствует заданию и установленным требованиям, имеется в наличии практическая часть ВКР, руководитель подписывает пояснительную записку, а также прилагает отзыв руководителя (Приложение Д). Текст ВКР

в обязательном порядке переплетается. Титульный лист оформляется в соответствии с приложением В. Оформление ВКР производится в соответствии с описанными ниже в данном методическом руководстве требованиями.

В случаях, когда руководитель считает, что работа студентом не выполнена в соответствии с заданием, результаты и цели не достигнуты даже на пороговом уровне, он пишет отрицательный отзыв о работе студента (не рекомендует работу к защите) и доводит его до заведующего выпускающей кафедрой. Пояснительная записка к ВКР в таком случае руководителем не подписывается, и студент к защите не допускается. В исключительных случаях решение о допуске к защите работ с отрицательными отзывами руководителя может быть принято на заседании выпускающей кафедры. К исключительным случаям относятся те случаи, когда все формальные требования студентом выполнены (имеется сброшюрованная ВКР, презентация, практическая часть ВКР), но он получил отрицательный отзыв руководителя.

2. ОПИСАНИЕ ПРОЦЕССА ЗАЩИТЫ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

2.1 Предварительная защита ВКР на выпускающей кафедре

За неделю до защиты в ГЭК выпускающей кафедрой назначается предварительная защита ВКР. Предварительная защита проходит перед комиссией, в которую входят преподаватели кафедры, назначенные приказом ректора (директора). Допускается участие иных заинтересованных преподавателей выпускающей или смежных кафедр. Для предварительной защиты обучающемуся необходимо иметь готовую ВКР и подписанный научным руководителем отзыв. Примерная форма представлена в приложении Г. В отзыве руководитель указывает, рекомендуется им или не рекомендуется ВКР к защите, а также характеризует и оценивает работу автора по четырехбалльной шкале (неудовлетворительно, удовлетворительно, хорошо, отлично).

В процессе предварительной защиты обучающийся кратко излагает суть выполненной работы и отвечает на вопросы членов комиссии. Студенту не нужно бояться процедуры предзащиты. **Цель предварительной защиты ВКР – помочь студенту адаптироваться к процедуре защиты, подготовить, отрепетировать доклад, презентацию, манеру, стиль изложения материала, сделать правильный акцент на выносимые на защиту результаты ВКР.** Также студент знакомится с примерным перечнем вопросов, которые ему могут задать во время прохождения ГИА, что дает ему (совместно с руководителем) возможность и время подготовить правильные аргументированные ответы.

2.2 Защита ВКР в Государственной экзаменационной комиссии

Выпускник представляет на защиту следующие документы:

Полностью оформленную ВКР, содержащую:

- стандартный титульный лист, подписанный выпускником, руководителем (консультантом), зав. выпускающей кафедры;
- текст ВКР с оглавлением, заключением, списком использованных источников и приложениями;
- акт проверки ВКР на антиплагиат в системе «Антиплагиат - ВУЗ»;
- отзыв руководителя (вкладывается в пояснительную записку);

Состав ГЭК и сроки публичной защиты утверждаются приказом ректора (директора филиала).

Рекомендуется следующий порядок публичной защиты (фактический порядок определяет председатель ГЭК):

1. Объявляется тема работы.

2. Выпускник в течение 7-10 минут излагает краткое содержание выполненной работы (текущее состояние рассматриваемого вопроса, методы и ход решения поставленных задач, и основные результаты, полученные в работе).

3. Выпускник отвечает на вопросы, предложенные членами ГЭК или присутствующими на защите (вопросы могут вытекать не только из конкретного содержания ВКР, но и из смежных областей, и носить как теоретический, так и прикладной характер).

4. Секретарем ГЭК зачитывается отзыв руководителя.

5. Выпускник произносит заключительное слово (по желанию).

После завершения всех запланированных на текущий день защит ГЭК выносит решение об оценке работы и защиты и о присвоении автору работы квалификации «бакалавр». Это делается на закрытом заседании ГЭК, работа которого не протоколируется. Как правило, решение принимается открытым голосованием. Решение ГЭК объявляется публично.

В случае уважительных причин, подтвержденных документально, приказом по Филиалу срок защиты ВКР может быть перенесен.

3. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

После согласования окончательного варианта выпускной квалификационной работы с руководителем ее распечатывают и переплетают в твердом переплете.

ВНИМАНИЕ! Выпускная квалификационная работа оформляется строго в соответствии с «ГОСТ 7.32-2017 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления (с Поправками)».

Авторы данного методического пособия не видят необходимости повторять содержание вышеприведенного ГОСТ. Им представляется, что компетенций выпускников вполне достаточно, чтобы самостоятельно ознакомиться с требованиями ГОСТ и выполнить их. Однако ряд важных моментов в оформлении ВКР представлены ниже.

В тексте выпускной квалификационной работы рекомендуется чаще применять красную строку, выделяя законченную мысль в самостоятельный абзац.

Слишком много цитат в работе приводить не следует, цитирование используется как прием аргументации. Частое цитирование приводит к повышению объема заимствованного текста, что отрицательно сказывается на проверке работы в системе «Антиплагиат - ВУЗ».

В выпускной квалификационной работе допускается использование материала из других источников, то есть можно излагать чужие мысли своими словами, но в этом случае необходимо делать ссылку на первоисточник.

В тексте выпускной квалификационной работы, кроме общепринятых буквенных аббревиатур, могут быть использованы вводимые лично авторами буквенные аббревиатуры, сокращенно обозначающие какие-либо понятия из соответствующих областей знаний. При этом первое упоминание таких аббревиатур указывается в круглых скобках после полного наименования, в дальнейшем они употребляются в тексте без расшифровки.

Список использованных источников включает в себя литературные, статистические и другие источники, материалы которых использовались при написании ВКР. Он состоит из таких источников, как монографическая и учебная литература, периодическая литература (статьи из журналов и газет), законодательные и инструктивные материалы, статистические сборники и другие отчетные и учетные материалы, Интернет-сайты. Порядок построения списка определяется автором выпускной квалификационной работы и научным руководителем.

Способы расположения материала в списке литературы должен быть хронологическим.

Последовательность брошюровки материала:

- **титульный лист;**
- **содержание;**
- термины и определения;
- перечень сокращений и обозначений;
- **введение;**
- **основная часть;**
- **заключение;**
- **список использованных источников;**
- приложения.

Обязательные структурные элементы выделены полужирным шрифтом. Остальные структурные элементы включают в ВКР по усмотрению руководителя.

В ВКР должны быть вложены (не сброшюрованы) отзыв руководителя и протокол проверки ВКР в системе «Антиплагиат - ВУЗ».

4. ТРЕБОВАНИЯ К РАЗДЕЛАМ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

К разделам выпускной квалификационной работы устанавливаются следующие смысловые требования.

4.1 Требования к содержанию раздела «Введение» выпускной квалификационной работы

Во введении обосновывается актуальность темы ВКР, определяются цель работы и её задачи, указывается объект исследования. Дается краткая характеристика содержания разделов ВКР, рассмотренных вопросов и полученных результатов.

Структурными элементами введения являются:

1. Актуальность работы.
2. Объект и предмет исследования.
3. Цель работы.
4. Задачи работы.
5. Новизна работы.
6. Практическая значимость работы.
7. Положения, выносимые на защиту.

4.2 Требования к содержанию раздела «Основная часть» выпускной квалификационной работы

Основная часть работы, как правило, состоит из трех глав. Каждая глава завершаться выводами. **Изменение структуры работы возможно только в исключительных случаях при согласии научного руководителя.**

В первой главе представляются результаты теоретического (мысленного) исследования выпускника по заявленной проблеме в теме работы (анализ предметной области).

Проводится обзор литературы и иных информационных источников по этой проблеме. При написании данной главы необходимо систематизировать материалы по теме ВКР и определить современное состояние изучаемой проблемы.

Материал исследования должен излагаться своими словами. При изложении материала необходимо указывать ссылки на соответствующие литературные источники. В необходимых случаях допускается использование цитат при соблюдении установленной формы цитирования. Анализ решений рекомендуется представлять в табличной форме с раскрытием характеристик оценки анализируемых объектов.

Выводы по главе должны отметить ожидаемый эффект от применения соответствующих методов, методик, инструментов и программного обеспечения.

Очень важно в первом разделе показать, что проблема, решаемая в ВКР на сегодняшний день актуальна, и требует детальной проработки.

Основным итогом данного раздела выпускной квалификационной работы должно явиться обоснование потребности в совершенствовании реализации решения выбранной задачи в условиях рассмотренного объекта.

Если говорить простыми словами, то выводы первого раздела должны отвечать на вопросы; **Что вы делаете?** и **Для чего вы это делаете.**

Во второй главе проводится анализ объекта исследования с позиций рассматриваемых задач ВКР. На основе имеющихся фактических материалов необходимо детально проанализировать состояние объекта, обязательно производится моделирование его бизнес-процессов. В этих условиях необходимо описать существующую практику решения поставленной задачи, обосновать выбор тех или иных инструментов, которые будут использованы при решении задач. Также оценивается экономическая, техническая и (или) социальная эффективность их внедрения в реальную информационную среду в области применения.

Важным является понимание автором ВКР, что он решает актуальную задачу предприятия (организации) при помощи современных информационных технологий.

Раздел должен начинаться изложением требований к решению рассматриваемой задачи. Постановка должна состоять из описания цели решения, входных и выходных данных, метода решения, а также ожидаемого экономического эффекта от решения поставленной задачи.

Так, если результатом выпускной квалификационной работы является разработка программного продукта или его модуля, базы данных, то в данной главе обучающийся проводит анализ современных методов и средств разработки программного обеспечения, применимых в конкретной предметной области, обосновывает выбор именно того средства, с помощью которого решается его задача. Также приводится обоснованный выбор языка программирования. Выбор инструментов разработки в ряде случаев может ограничен, ввиду того, что на предприятии, для которого студент выполняет работу, уже сложилась практика использования определенных инструментов и средств разработки, СУБД и т. п. В данном случае обучающийся также должен сделать соответствующее обоснование.

При этом совершенно неправильным будет подход, при котором обучающийся сделал великолепный обзор инструментов разработки, но ставит перед фактом, что он выбрал такой-то инструментарий, потому что он так решил, или потому что он знаком только с ним. Должны четко быть сформулированы критерии выбора, например: доступность (бесплатное программное обеспечение),

популярность, наличие поддержки мировым сообществом (множество библиотек с открытым кодом), кроссплатформенность и т. д. Желательно все критерии свести в таблицу, дать оценку каждому и сделать обоснованный выбор того продукта, который наберет большее количество баллов в вашей оценке.

В любом случае, обоснование выбора инструментов должно быть убедительным.

Основным итогом данной главы должен быть ответ на вопрос: **Какие инструменты я использую для решения поставленных задач и почему именно такие?**

В третьей главе излагается практическая реализация задач выпускной квалификационной работы. Приводятся архитектура и структура информационной системы (технического устройства, прототипа, сети передачи данных и т. п.), используемой СУБД, алгоритмы решения задач, другие важные аспекты практического использования ВКР. Например, при разработке или доработке информационной системы обязательно приводится описание ее **архитектуры**, описание ее программной реализации если информационная система разработана «с нуля» или доработки существующей. Студент подробно описывает проделанную конкретно им работу, сопровождая описание рисунками, схемами, блок-схемами алгоритмов, используя стандартные нотации IDEF0, UML и др.

В данной главе так же рассматриваются сценарии работы пользователя с системой с подробными описаниями и примерами экранных форм, с которыми работает пользователь. Необходимо рассмотреть вопросы информационной безопасности при проектировании и будущей эксплуатации информационной системы (информационной сети, базы данных и т.п.). Также рекомендуется рассмотреть способы и методы тестирования созданной (модифицированной) информационной системы (базы данных, вычислительной сети, электронного устройства и т. п.)

Не следует перегружать главу исходными текстами программ, второстепенными схемами, рекомендуется помещать их в приложения с соответствующими ссылками на них.

При написании данной главы автор должен понимать, что здесь он **подробно расписывает результаты выполнения поставленных задач.**

Не забывайте про краткие выводы по каждой главе!

Пример оформления пояснительной записки приведен в Приложении Е.

4.3 Требования к содержанию раздела «Заключение» выпускной квалификационной работы

В заключении обобщаются результаты, полученные во всех главах, делаются общие для всей работы выводы о достижении поставленной цели работы, полученных выгодах и показателях эффективности.

4.4 Требования к содержанию раздела «Список использованных источников» выпускной квалификационной работы

В список использованных источников включаются ссылки на литературные и электронные источники, использованные при написании работы, нормативные документы и материалы организаций, использованные при подготовке выпускной квалификационной работы, технологические регламенты и стандарты эксплуатации информационных систем.

4.5 Требования к содержанию раздела «Приложения» выпускной квалификационной работы

В приложения выносятся дополнительный материал, необходимый для подтверждения рассматриваемых положений. Приложениями могут содержать: графический материал, таблицы большого формата, расчеты, описания алгоритмов и т. д. Первым приложением в пояснительной записке (Приложение А) должно быть техническое задание на ВКР.

4.6 Требования к объему выпускной квалификационной работы

Объем работы не должен превышать 60–70 машинописных страниц, (включая таблицы, рисунки и графики), набранных 14 шрифтом Times New Roman с межстрочным интервалом 1,5 пункта.

5. РЕКОМЕНДАЦИИ ПО СТРУКТУРЕ КОМПЬЮТЕРНОЙ ПРЕЗЕНТАЦИИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Как правило, при защите докладчику отводится на выступление 8-10 минут. Стандартом является построение доклада на основе компьютерной презентации (MS Power Point или аналог). Рекомендуется следующий порядок слайдов:

1. Название темы, реквизиты автора и научного руководителя.
2. Цель, задачи, методы (1-2 слайда).
3. Краткая характеристика объекта (название, род деятельности). (1 слайд)
4. Модели управления бизнес-процессами подлежащие оптимизации (1-2 слайда).
5. Описание предлагаемого решения, позволяющего повысить эффективность бизнес-процессов предприятия / организации (10-12 слайдов).
6. Заключение, наиболее важные выводы (1 слайд).

Не рекомендуется читать текст с экрана.

Не рекомендуется использование сложных мультипликационных презентаций, допустимо использование коротких видеовставок, иллюстрирующих работу программного продукта или устройства.

Следует внимательно подходить к цветовой палитре презентации, размеру шрифта текста – текст должен быть контрастным и хорошо читаться на экране проектора. Все приводимые рисунки, схемы, графики должны быть читаемы. Если, например, структура вашей базы данных настолько большая, что на слайде выходит очень мелко – разбейте ее на несколько слайдов, или оставьте только важные ее элементы.

Если по результатам работы предприятие (организация), для которой вы осуществляли разработку, составила акт внедрения, отзыв, благодарственное письмо – не стесняйтесь и покажите это комиссии на одном из заключительных слайдов.

ЗАКЛЮЧЕНИЕ

В данном методическом пособии рассмотрены основные аспекты работы над выпускной квалификационной работой, изложен порядок ее выполнения, прохождения смотров, нормоконтроля, предзащиты и самой процедуры защиты. Перечислен состав документов, которые студент должен оформить в процессе работы над ВКР, изложены правила их оформления и требования к содержательной части. Приведены шаблоны всех необходимых документов в приложениях. Авторы надеются, что данное методическое пособие поможет студенту быстрее и эффективнее пройти процедуру государственной итоговой аттестации.

Приложение А

Примерные темы выпускных квалификационных работ

1. Проектирование автоматизации группы бизнес-процессов предприятия.
2. Разработка автоматизированной системы автоматической фиксации номеров железнодорожных вагонов с использованием компьютерного зрения.
3. Разработка веб-интерфейса для модуля компьютерного зрения.
4. Разработка мобильного приложения для конфигурирования модуля компьютерного зрения.
5. Разработка мобильного приложения для решения задачи автоматической фиксации посещаемости занятий студентами.
6. Разработка серверной части приложения для решения задачи автоматической фиксации посещаемости занятий студентами.
7. Разработка подсистемы распознавания лиц для проекта автоматической фиксации посещаемости занятий студентами.
8. Разработка мобильного приложения для управления микроконтроллерами систем централизованного отопления.
9. Разработка онлайн сервиса автоматизированного расчета стоимости работ по лазерной/плазменной/гидроабразивной резке материалов на основе загружаемого векторного чертежа.
10. Разработка система распознавания образов для определения качества изготовления детских сидений на роботизированных комплексах.
11. Разработка интерактивного голосового помощника для приемной комиссии филиала.
12. Разработка интерактивного голосового помощника онлайн сервиса (по видам экономической деятельности).
13. Разработка чат-бота для приемной комиссии филиала.
14. Разработка чат-бота онлайн сервиса (по видам экономической деятельности).
15. Организация продвижения и эксплуатации Интернет-проекта.
16. Проектирование Интернет-магазина.
17. Автоматизация решения задачи финансового аналитика на предприятии.
18. Внедрение многоуровневой архитектуры «тонкого клиента» в информационной сети компании.
19. Внедрение информационной системы управления логистикой и складом предприятия.
20. Проектирование хранилищ данных BigData и аналитической системы предприятия.
21. Управление проектом подготовки и продвижения социального веб-сервиса.
22. Разработка приложения для поддержки принятия маркетинговых решений на предприятии на основе методов DataMining.

23. Внедрение системы управления взаимодействия с поставщиками.
24. Проектирование сервисной архитектуры информационной системы предприятия.
25. Автоматизация бизнес-процесса закупочной деятельности предприятия.
26. Автоматизированная информационная система поддержки маркетинговой деятельности субъекта малого предпринимательства.
27. Внедрение автоматизированной информационной системы для учета работы с клиентами предприятия.
28. Особенности управления системой электронного бизнеса (ЭБ) (на конкретном примере).
29. Создание виртуального предприятия (структура, тип, развитие) (для конкретных применений).
30. Создание виртуальной торговой площадки (структура, тип, развитие) (для конкретных применений).
31. Автоматизация процессов складской логистики в производственной компании.
32. Автоматизация процессов службы ИТ-поддержки компании.
33. Информационная поддержка процесса сопровождения клиентов в интернет-магазине компании.
34. Повышение уровня обслуживания клиентов на основе автоматизации процесса управления заказами.
35. Разработка инструментария формирования мобильной отчетности.
36. Разработка интерактивной отчетности на основе информационного поля компании.
37. Совершенствование информационной поддержки торговой компании на основе применения интернет-технологий.
38. Совершенствование процесса формирования договоров на основе внедрения системы электронного документооборота.
39. Организация разработки корпоративного информационного портала (его структура и механизмы функционирования) для конкретных применений.
40. Исследование проблемы безопасности в системах ЭБ и способы ее решения (для конкретных применений).
41. Разработка модуля онлайн платформы управления предприятием (на конкретном предприятии).
42. Автоматизация решения задач маркетолога в страховой компании.
43. Разработка информационной системы управления персоналом на предприятии сферы информационных технологий.
44. Внедрение системы управления взаимодействия с клиентами.
45. Проектирование автоматизированной системы ИТ отдела крупного предприятия.

46. Разработка комплексной автоматизированной системы предприятия с использованием технологий 1С.
47. Проектирование стратегических бизнес-процессов предприятия.
48. Использование систем больших данных в управлении предприятием.

Приложение Б

Образец заявления о разрешении выполнения темы выпускной квалификационной работы

Заведующему кафедрой

от обучающегося курса

Профиль подготовки

ЗАЯВЛЕНИЕ

Прошу Вас разрешить мне выполнять выпускную квалификационную работу
на тему:

и назначить научного руководителя _____

Дата _____

Подпись _____

Согласовано _____ ФИО руководителя

Приложение В

Образец титульного листа на выпускную квалификационную работу

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВА-
ТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ «УД-
МУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» В ГОРОДЕ ВОТКИН-
СКЕ
(ФИЛИАЛ ФГБОУ ВО «УДГУ» В Г. ВОТКИНСКЕ)

Допустить к защите _____

Зав.кафедрой _____

«__» _____ 20__ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по направлению **09.03.03 «Прикладная информатика»**

подготовки

профиль **«Прикладная информатика в экономике»**

на тему: _____

Обучающийся группы _____

фамилия, имя, отчество и подпись

Научный руководитель _____

ученое звание, ученая степень, ФИО и подпись

Воткинск 20__

Приложение Г

Образец задания на выпускную квалификационную работу

ФИЛИАЛ ФГБОУ ВО «УДГУ» В Г. ВОТКИНСКЕ

Кафедра «_____»

Утверждаю:

Зав. кафедрой _____

«__» _____ 202__ г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Обучающемуся _____

на тему _____

Тема утверждена приказом по Филиалу ФГБОУ ВО «УДГУ» В Г. ВОТКИНСКЕ
«_____» _____ 202__ г. № _____

Срок сдачи обучающимся работы «_» _____ 202__ г.

Исходные данные к работе _____

Содержание расчетно-пояснительной записки (перечень подлежащих раз-
работке вопросов):

1. _____
2. _____
3. _____
4. _____

Подпись обучающегося _____

Подпись научного руководителя _____

Приложение Д

Образец отзыва руководителя выпускной квалификационной работы

ОТЗЫВ

на выпускную квалификационную работу обучающегося

на тему _____

Выпускная квалификационная работа обучающегося _____ выполнена на страницах, состоит из введения, _____ глав, заключения, списка использованных источников из ____ наименований и приложения.

Далее в отзыве указывается актуальность темы исследования, дается краткая характеристика работы выпускника при написании ВКР, его добросовестность, аккуратность и степень подготовленности как специалиста.

В отзыве указывается, имеет ли обучающийся достаточную теоретическую подготовленность, умения и знания при решении практических задач.

В заключительной части дается общая оценка выпускной квалификационной работы и возможность допуска ее к защите.

_____ ФИО руководителя ВКР

Приложение Е

Пример оформления пояснительной записки

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» В ГОРОДЕ ВОТКИНСКЕ
(ФИЛИАЛ ФГБОУ ВО «УДГУ» В Г. ВОТКИНСКЕ)

Допустить к защите _____

Зав.кафедрой _____

«__» _____ 20__ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по направлению

09.03.03 «Прикладная информатика»

подготовки

профиль

«Прикладная информатика в экономике»

на тему: «Разработка приложения для автоматизации процесса подготовки листов изменений заводской технической документации ООО «СЧЕТМАШ»

Обучающийся группы ОБ-Вт-090303-41 Иванов Иван Иванович

фамилия, имя, отчество и подпись

Научный руководитель к. т. н., доцент Петров Петр Петрович

ученое звание, ученая степень, ФИО и подпись

Воткинск 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Анализ предприятия и предметной области	7
1.1 Описание организации.....	7
1.2 Описание инструментария проектировщика.....	8
...	
1.4 Описание предметной области	12
Выводы	13
2 Анализ бизнес-процессов предприятия, выбор и обоснование инструментов разработки	15
2.1 Анализ средств прототипирования и моделирования.....	15
2.2 Анализ бизнес-процессов подготовки заводской технической документации к выдаче потребителю	17
2.3 Анализ бизнес-процессов подготовки листов изменений заводской технической документации.....	19
2.4 Анализ инструментов разработки приложений для персонального компьютера.....	20
Выводы	29
3 Проектирование информационной системы	31
3.1 Построение логики приложения.....	31
3.2 Проектирование структуры базы данных.....	34
3.3 Прототипирование интерфейса приложения	37
3.4 Создание базы данных	42
3.5 Разработка интерфейсной части приложения	46
3.6 Разработка программной части приложения.....	48
3.1.2 Создание диаграммы БД средствами MS SQL Server Management Studio.....	47
3.2 Разработка интерфейсной части приложения.....	49
3.3 Разработка программной части приложения.....	54
Выводы	56
ЗАКЛЮЧЕНИЕ	57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	58
ПРИЛОЖЕНИЯ	60

ВВЕДЕНИЕ

В наши дни информационные технологии развиваются быстрыми темпами, они охватывают все больше сфер жизни людей и становятся неотъемлемой частью любого вида деятельности. Ранее процесс обработки информации осуществлялся преимущественно вручную, используя бумажные носители это приводило к трате большого количества времени, ресурсов и в целом замедляло работу производства. Постепенно объем информации начал увеличиваться, что привело к возникновению проблем с ее обработкой. С появлением компьютеров появилась возможность автоматизации данного процесса. Автоматизация влечет за собой эффективность в том виде деятельности, в котором используется: автоматизация процессов в работе бухгалтерии за счет сокращения количества бумажной документации, создании отчетов; автоматизация процессов производства за счет роботизации ручного труда и многое другое.

Данная работа основана на исследовании бизнес-процессов предприятия ООО «СЧЕТМАШ», основным направлением деятельности которого является производство широкого спектра электрооборудования и комплексное решение задач по строительству, реконструкции и модернизации систем распределения и управления электроэнергией. Данное предприятие работает с большим объемом информации, соответственно время на оформление бумажной документации занимает довольно большую часть рабочего времени, например, проектного отдела. Таким образом, была поставлена задача разработать приложение для автоматизации процесса подготовки листов изменений заводской технической документации, для повышения эффективности бизнес-процесса взаимодействия проектного отдела с заказчиком продукции завода, проектным институтом, заводом по производству блочных модульных зданий и самим производством ООО «СЧЕТМАШ». Поэтому данная тема ВКР является актуальной.

Объектом исследования данной работы выступают бизнес-процессы создания сотрудником проектного отдела организации ООО «СЧЕТМАШ» заводской технической документации.

А в роли **предмета** исследования выбраны бизнес-процессы подготовки листов изменений для заводской технической документации организации ООО «СЧЕТМАШ».

Выбор **объекта и предмета** исследования неслучаен, так как повышение производственной эффективности проектного отдела, который выступает одним из основных бизнес-процессов работы данного предприятия, играет существенную роль в повышении уровня эффективности деятельности самого предприятия.

Целью данного дипломного проекта является: повышение эффективности процесса подготовки листов изменений заводской технической документации ООО «СЧЕТМАШ» за счет разработки программы, автоматизирующей этот процесс.

В рамках поставленной цели ВКР решаются следующие задачи:

1. Исследование бизнес-процессов создания сотрудником проектного отдела организации заводской технической документации;
2. Исследование предметной области;
3. Исследование инструментов разработки приложений для персонального компьютера;
4. Написание технического задания на разработку приложения;
5. Прототипирование интерфейса;
6. Проектирование базы данных;
7. Разработка автоматизированной системы;
8. Описание созданной автоматизированной системы.

Структура настоящей ВКР:

1. Глава 1 – Современные инструменты разработки приложений для персонального компьютера.

В главе анализируются бизнес-процессы проектного отдела предприятия, бизнес-процессы создания сотрудником проектного отдела заводской технической документации и предметная область, для которой ведется разработка. Выполняется обзор инструментов для моделирования интерфейсной и программирования программно-аппаратной части приложения.

2. Глава 2 – Проектирование и прототипирование

В главе описана модель базы данных, структура классов приложения, а также этапы и результаты прототипирования интерфейса приложения.

3. Глава 3 – Разработка автоматизированной системы

В главе представлены основные аспекты разработки интерфейса и приложения для автоматизации процесса подготовки листов изменений заводской технической документации.

4. Заключение – представлены основные выводы по исследованиям в результате выполнения ВКР, приводятся практические результаты работы.

1 АНАЛИЗ ПРЕДПРИЯТИЯ И ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание организации

ООО «СЧЕТМАШ» (далее – предприятие) специализируется на производстве электрооборудования от 0,4 до 35 кВ., комплексном решении задач по реконструкции, строительству и модернизации систем распределения и управления электроэнергией. Благодаря развитой дилерской сети, поставки и сопровождение заказов осуществляется по всей территории России, Белоруссии и Казахстана. Начало производства КРУ "Классика" стартовало в апреле 2003 года в городе Санкт-Петербург. В развитии долгосрочного совместного проекта Российской группы компаний "Таврида Электрик" и польской компании "ELEKTROBUDOWA S. A." сделан следующий шаг – строительство завода с полным циклом производства [1].

Результаты работы коллектива завода в области менеджмента качества подтверждены ежегодными положительными заключениями внешних аудиторских организаций и выдачей сертификата на соответствие требованиям ГОСТ Р ISO 9001-2015 (ISO 9001:2015).

В середине 2009 года в г. Воткинск Удмуртской Республики в эксплуатацию был введен один из самых современных российских заводов, занимающийся производством КРУ – комплектных распределительных устройств и КТП – комплектных трансформаторных подстанций среднего напряжения. С начала производственной деятельности было выполнено порядка 2000 заказов по индивидуальным проектам разной степени сложности и отгружено заказчикам различных сфер деятельности от Владивостока до Калининграда и от Полярного круга до Казахстана свыше 20 тысяч шкафов КРУ.

На отечественном рынке продукция ООО «СЧЕТМАШ» известна уже давно и зарекомендовала себя как надежное современное оборудование высокого качества [1].

Проектный отдел предприятия, является важнейшим этапом производства, именно он готовит проект: электросхемы, чертежи, документацию. Его

основной задачей является поддержка проекта в актуальном состоянии на протяжении всего жизненного цикла продукции, соблюдение требований заказчика и составляемой технической спецификации. Организация эффективной работы именно этого отдела имеет большое влияние на другие этапы производства [1].

1.2 Описание инструментария проектировщика

Основным инструментом работы сотрудника проектного отдела является система автоматизированного проектирования (САПР) *E3.Series* компании *Zuken* поддержкой этого продукта на территории России занимается компания *Point*.

E3.Series – программный комплекс, позволяющий решить полный цикл проектных задач в области проектирования систем электротехники, АСУ ТП, КИПиА – начиная от создания функциональных схем, заканчивая чертежами компоновки шкафов и панелей, а также выпуском конструкторской документации. Это третье поколение ECAD-систем, позволяющее реализовать требования, предъявляемые к современным проектам и технологиям автоматизации производственных процессов. *E3.series* служит платформой для взаимодействия инженеров разных специальностей. Она включает средства работы с базой данных, графическим редактором и генератором отчетов. Разработка электрических, пневматических, гидравлических, технологических схем, а также схем автоматизации выполняется в единой среде проектирования и по единым правилам. Система *E3.series* создана германской компанией *CIM-Team*, входящей в международный холдинг *Zuken* [2].

Автоматизированная система управления технологическим процессом (АСУ ТП) – группа решений технических и программных средств, предназначенных для автоматизации управления Технологическим Процессом. Может иметь связь с более общей автоматизированной системой управления предприятием [2].

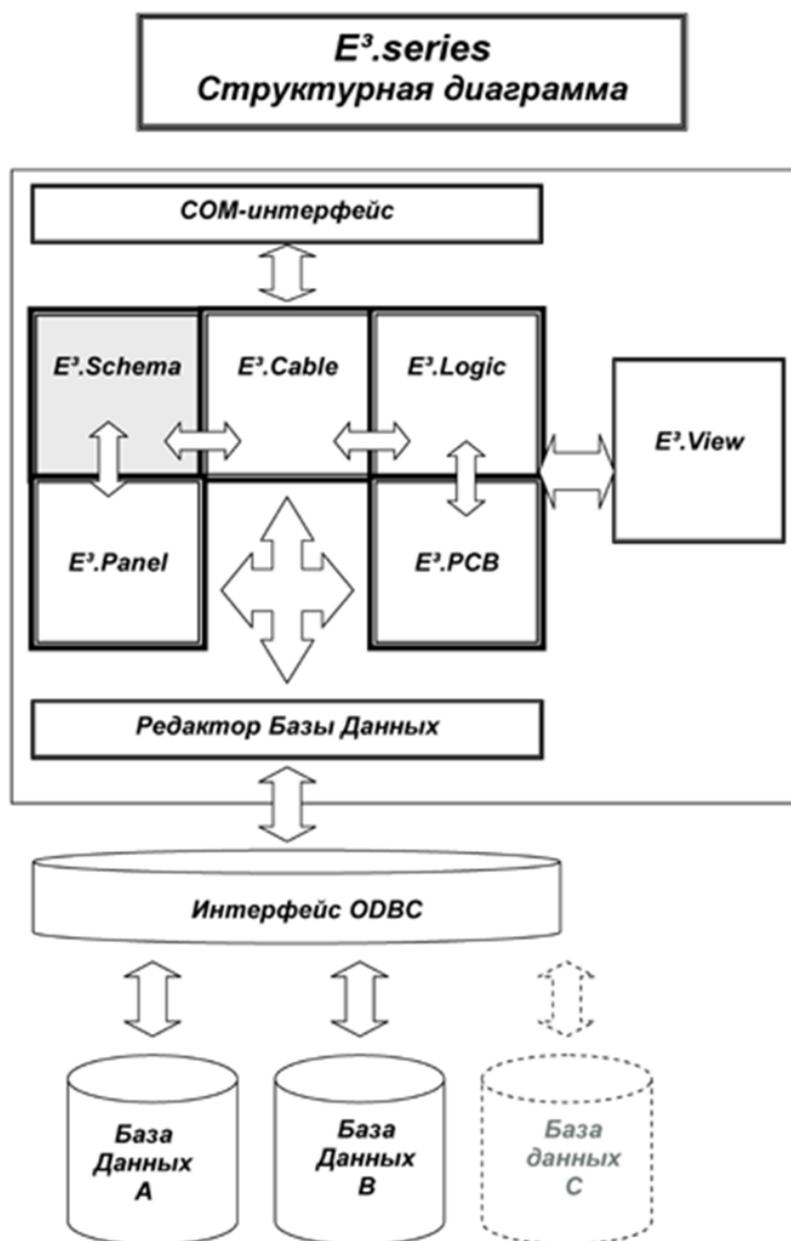


Рисунок 1 – Структурная диаграмма программного комплекса E3.Series

Стандартные модули

Пакет приложений E3.series[2] состоит из отдельных модулей, построенных на едином графическом программном ядре. Такая архитектура обеспечивает совместное использование одних и тех же данных всеми модулями пакета без дополнительного конвертирования. В стандартную поставку входят такие модули как:

E3.Schematic

В основе платформы E3.series лежит модуль E3.schematic, предназначенный для разработки всех типов принципиальных схем. Графический редактор

модуля позволяет осуществлять быстрый поиск нужного изделия в базе данных и размещать его на поле чертежа.

Модуль E3.schematic – это средство проектирования различных типов схем, начиная от технологической схемы, схемы автоматизации и заканчивая схемой внешних подключений или схемой внешних проводок, а также генерировать комплект выходной документации. Причём, перечень необходимых схем и требуемой документации определяет сам пользователь. Так как на этапе создания базы данных пользователь указывает для каждого типа схемы «своё» УГО, то при добавлении изделия на тот или иной тип схемы, система автоматически вынесет на неё соответствующий символ [3].

E3.Cable

Модуль E3.Cable позволяет проектировать кабельные и жгутовые связи с использованием как функций работы с принципиальными схемами, так и специальных функций работы с логическими блоками – «черными ящиками». Задание взаимосвязей между блоками легко и эффективно осуществляется динамически или с использованием типовых проводов, кабелей и жгутов, описанных в базе данных изделий. Каждый объект (разъем, жгут или кабель) может быть графически представлен в виде, соответствующем типу схемы или чертежа. Используя множество видов представления объектов на различных листах проекта, E3.Cable позволяет создавать – в дополнение к блок-схемам – динамические чертежи жгутов, таблиц цепей и сигналов. После модификации любого из видов (графических изображений) объекта все другие виды модифицированного объекта автоматически изменяются [3].

E3.Panel

Основная задача модуля E3.Panel – предоставить проектировщику быстрый и точный инструмент по созданию необходимого комплекта конструкторской документации. Чтобы скомпоновать, например, силовой шкаф, достаточно вызвать подходящую конструкцию как изделие из базы данных и выполнять компоновочный чертеж на развертке шкафа. Перед началом компоновки шкафа модуль E3.Panel рассматривает атрибуты всех изделий из проекта, выбирает

только те изделия, которые предполагается разместить в этом шкафу, и помещает их в специальный раздел проекта «Неразмещенные изделия компоновки». После того, как в процессе компоновки изделие размещается в шкафу, оно автоматически переносится в раздел «Размещенные изделия компоновки».

E3.Viewer

Свободно распространяемый автономный модуль для просмотра проектов E3.series (без необходимости установки полнофункциональной версии E3.series). Работает как с локальными проектами, так и с многопользовательскими. Позволяет проекты выводить на печать.

Многопользовательский режим

- Группа пользователей работает одновременно над одним проектом
- Возможности администрирования проекта
- Архитектура «Клиент – Сервер»
- Сервер имеет прямой доступ к проекту
- Клиент связывается с проектом через сервер
- Сервер работает с базой данных
- Microsoft SQL Server
- Oracle
- Коммуникация через TCP/IP порты
- Клиент может работать в одном из режимов: «однопользовательский» или «многопользовательский» [2].

APM проектировщика – инструмент, созданный на предприятии, как надстройка для E3.Series, который взаимодействует с открытым проектом средствами COM соединения, для получения желаемого расширения функционала САПР и автоматизации некоторых процессов.

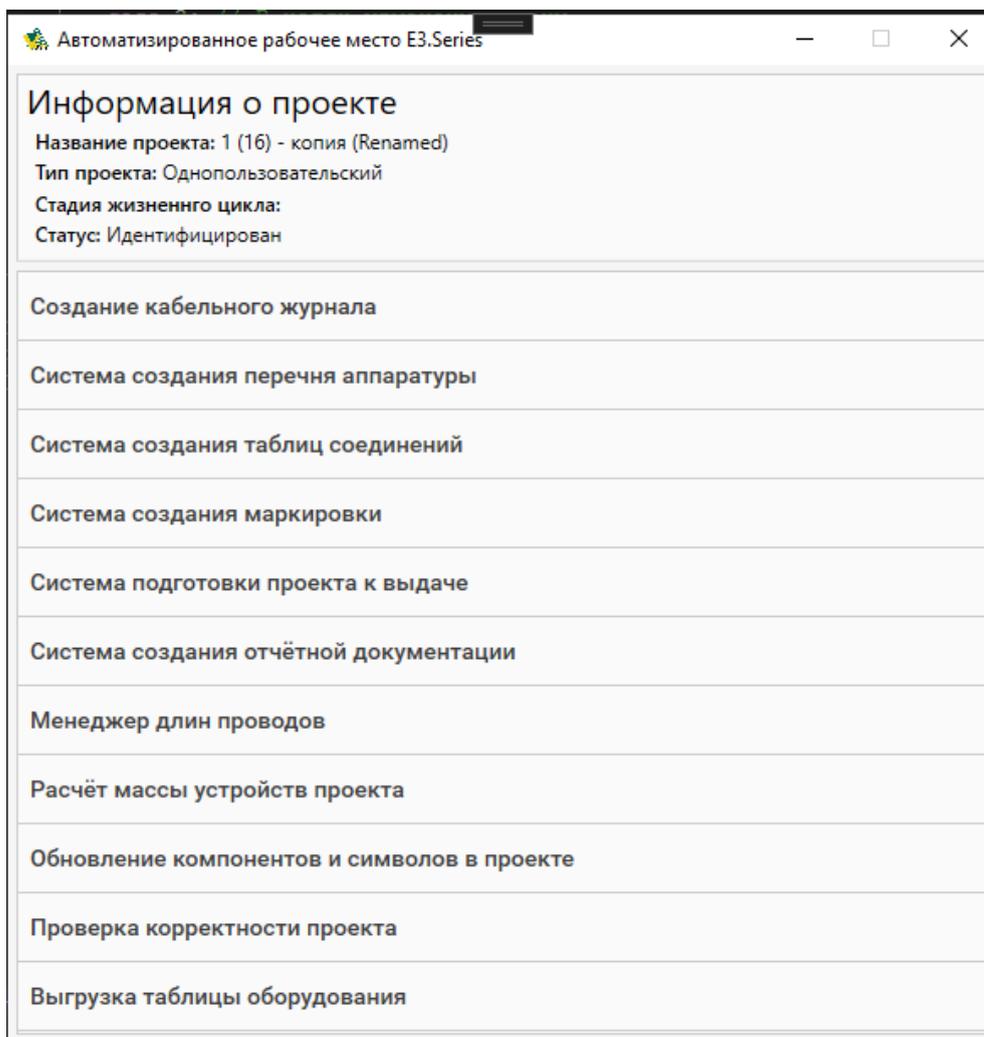


Рисунок 2 – АРМ проектировщика

• • •

1.4 Описание предметной области

После этапа проектирования идет формирование заводской технической документации – документация на продукцию предприятия, туда входят:

- Таблица оборудования;
- Лист примечаний (описание аббревиатур и сокращений);
- Кабельный журнал;
- Перечень аппаратуры;
- Таблица соединений;
- План шинок;
- Информационная сеть;

- Таблицы подключения клеммных рядов;
- Чертежи общего вида;
- Схемы электрические-принципиальные;
- Листы изменений.

Потребителями данной документации являются производство предприятия и заказчик, а также проектные институты. Данные потребители на протяжении всего жизненного цикла разработки нуждаются в листах изменений, для более быстрого и понятного изучения и сравнения двух версий ЗТД.

Листы изменений – сводные таблицы различий предыдущей и текущей версией заводской технической документации.

Заводская техническая документация – полная документация по описанию производимого продукта с полным набором сводных таблиц по вышперечисленным объектам оформленная по стандартам предприятия.

В данный момент подготовка листов изменений осуществляется вручную, без использования должных средств автоматизации. Это влечет, в свою очередь, большие трудозатраты. Соответственно, требуется организовать, автоматизировать процесс подготовки листов изменений ЗТД к выдаче потребителю.

Выводы

В результате анализа полученной информации и систематизации её можно сделать следующие выводы: заказчиком разрабатываемой продукции является проектный отдел ООО «СЧЕТМАШ», а конечным потребителем – сотрудник проектного отдела – проектировщик. Основными инструментами являются САПР E3.Series и АРМ проектировщика.

Предметной областью является документация на продукцию предприятия. Потребителями данной документации являются производство предприятия и заказчик, а также проектные институты. Данные потребители на протяжении всего жизненного цикла разработки нуждаются в листах изменений, для более быстрого и понятного изучения и сравнения двух версий ЗТД.

В данный момент подготовка листов изменений осуществляется вручную,

без использования должных средств автоматизации. Это влечет, в свою очередь, большие трудозатраты. Соответственно, требуется повысить эффективность процесса подготовки листов изменений ЗТД к выдаче потребителю. Это необходимо автоматизировать ряд бизнес-процессов предприятия.

2 АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ, ВЫБОР И ОБОСНОВАНИЕ ИНСТРУМЕНТОВ РАЗРАБОТКИ

2.1 Анализ средств прототипирования и моделирования

На сегодняшний день существует огромное количество инструментария для разработки моделей бизнеса, они используют различные языки моделирования, но все это можно выделить в три подхода, два из которых (функциональный и процессный) стандартизированы и широко применяются:

- Функциональный;
- Процессный;
- Ментальный.

В рамках данной работы будут рассмотрены функциональный и процессный подходы.

Функциональное моделирование

Рассматривает бизнес как функцию, своего рода «черный ящик». Для данной модели проектирования важны не сами процессы низкого уровня, а сама результативность отдельной функции. Функции имеют точку входа – объекты обработки и точку выхода – результат действия. Например, точкой входа может служить проект, как обрабатываемая информация, а выхода заводская техническая документация [12].

Данные модели также можно использовать для демонстрации своих идей и вариантов решений, двигаться от общего к деталям, по мере необходимости разделять и декомпозировать функции.

Одной из самых популярных нотаций является *IDEF0 (ICAM Definition)* основана на следующих концептуальных положениях:

- Модель – искусственный объект, представляющий собой отображение;
- (образ) системы и ее компонентов;
- Блочное моделирование и его графическое представление;
- Лаконичность и точность;
- Передача информации;

- Строгость и формализм;
- Итеративное моделирование;
- Отделение «организации» от «функций».

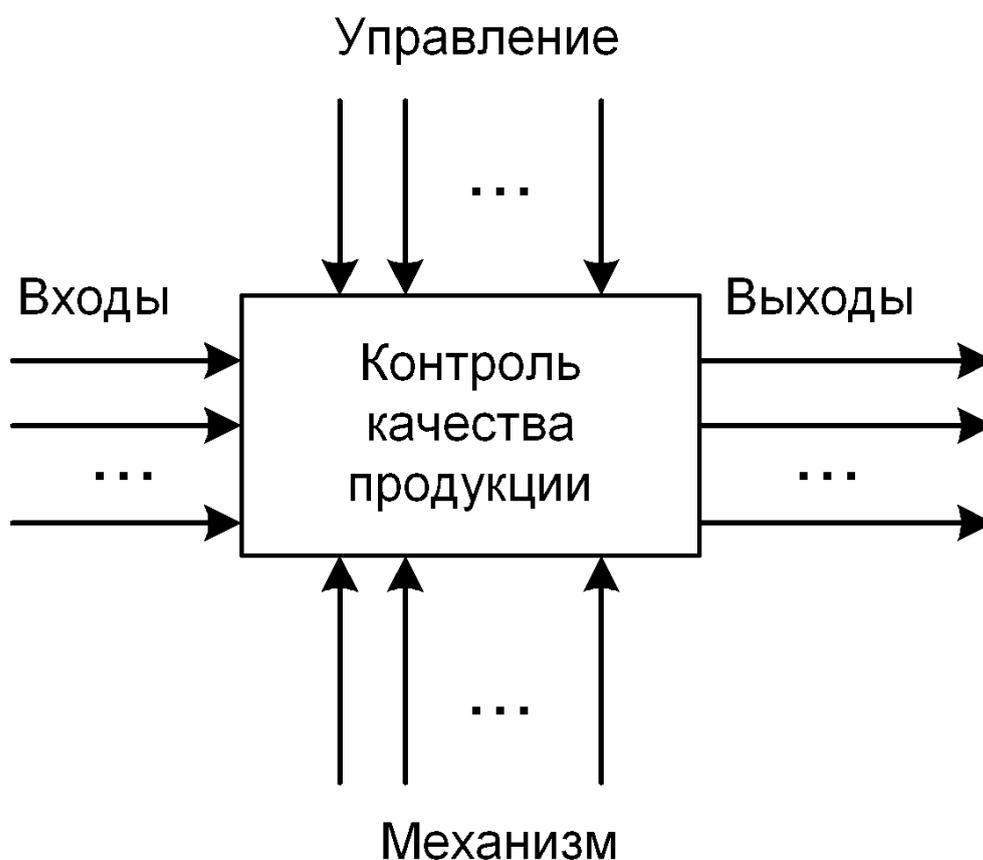


Рисунок 3 – Пример функциональной модели нотации *IDEF0* [12]

Процессное моделирование

Рассматривается с точки зрения нотации BPMN, как одного из наиболее распространенных стандартов процессного моделирования.

Процесс с точки зрения бизнес-модели – это последовательность каких-то событий и действий, которые имеют начало и конец.

В этом кроется основное отличие процессного моделирования от функционального. Функциональное моделирование рассматривает бизнес-модель с точки зрения входа и выхода (имеющихся ресурсов и желаемого результата). А процессное основано на последовательности действий в определенных границах, в случае BPMN это будут начало и конец события [12].

Все процессы могут разбиваться (детализироваться) на подпроцессы

вплоть до детализации на уровне задач, то есть действий, дальнейшая детализация которых невозможна. Процесс – это последовательность действий, которую необходимо выполнить, чтобы получить определенный результат. Необходимо отметить, что в модели бизнеса как процесса результат может и не быть явным в отличие от функциональной модели [12].

Принципиальное отличие процессного моделирования от функционального заключается в том, что при процессном моделировании основное внимание отдается не результату, а тому, что нужно сделать для получения результата.

В данной работе, для графического представления моделей бизнес-процессов будут использоваться именно рассмотренные нотации.

2.2 Анализ бизнес-процессов подготовки заводской технической документации к выдаче потребителю

В результате анализа работы проектного отдела, а именно работы по созданию ЗТД и подготовки к выдаче, было выделено несколько этапов работы, которую могут делать один или несколько сотрудников:

- Этап проектирования. На данном этапе производится проектирование продукции на основании технической спецификации заказа, своего рода «техническое задание» для проектного отдела;
- Этап тестирования. На данном этапе производится проверка проекта на ошибки различного характера приложением «АРМ проектанта»;
- Этап подготовки дополнительных листов. На данном этапе производится формирование с помощью «АРМ проектанта» документация по описанию отдельных блоков проекта: кабели, соединения, аппаратура и оборудование с последующим созданием соответствующих журналов;
- Этап формирования ЗТД. Этап автоматизирован не полноценно – с помощью приложения «АРМ проектанта» в проекте формируются дополнительные разделы «Тома», где все полученные ранее журналы, чертежи проекта будут структурированы;
- Этап формирования листов изменений. Этот этап требует тщательной про-

работки и автоматизации. Так как заказчик и производство предприятия нуждаются на протяжении всего жизненного цикла проекта получать различия между версиями. Текущее состояние: работа осуществлена (частично) в программе *MS Excel* только над объектами соединений, без дополнительной информации, которую невозможно обработать средствами *Excel*.

Исходя из этой информации, была построена модель бизнес-процессов на основе нотации *IDEF0*.

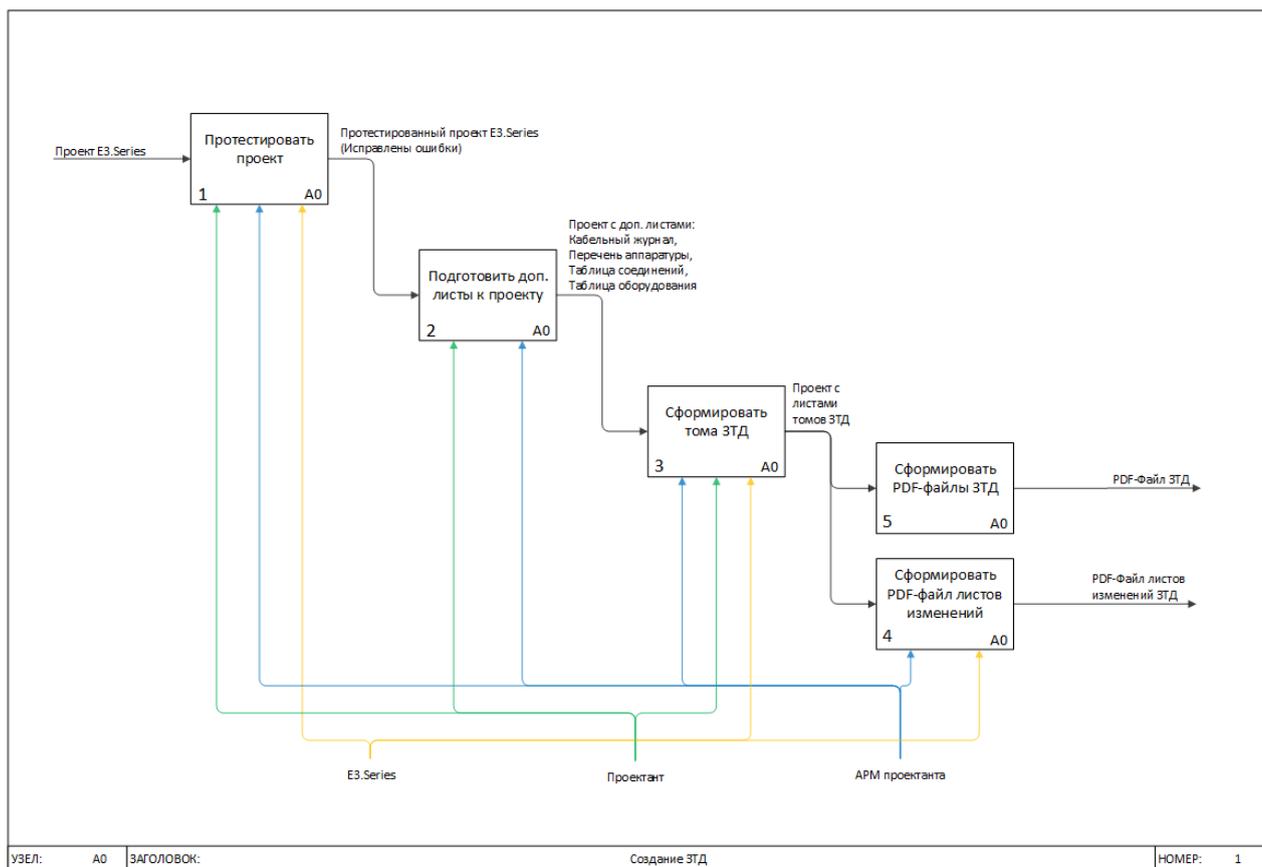


Рисунок 4 – Бизнес-процессы подготовки ЗТД к выдаче потребителю

Далее будет расписана модель будущих бизнес-процессов по подготовке листов изменений ЗТД.

2.3 Анализ бизнес-процессов подготовки листов изменений заводской технической документации

В соответствии с техническим заданием заказчик (проектный отдел предприятия) требует организовать, автоматизировать процесс подготовки листов изменений ЗТД к выдаче потребителю.

После анализа технического задания заказчика, были составлены следующие выводы:

- Прослеживаются четкие этапы процесса;
- Имеются дополнительные объекты отслеживания изменений;
- Требуется слияние процесса подготовки самого ЗТД и листов изменений.
- Исходя из этого можно выделить следующие этапы процесса:
- Выбор потребителя. На данном этапе производится выбор потребителя изменений;
- Анализ изменений. На данном этапе производится анализ полученных данных на предмет ошибок;
- Подготовка *PDF*-файлов. На данном этапе производится подготовка документации к выдаче.

Составлена схема бизнес-процессов на основании выводов и технического задания заказчика (проектного отдела предприятия).

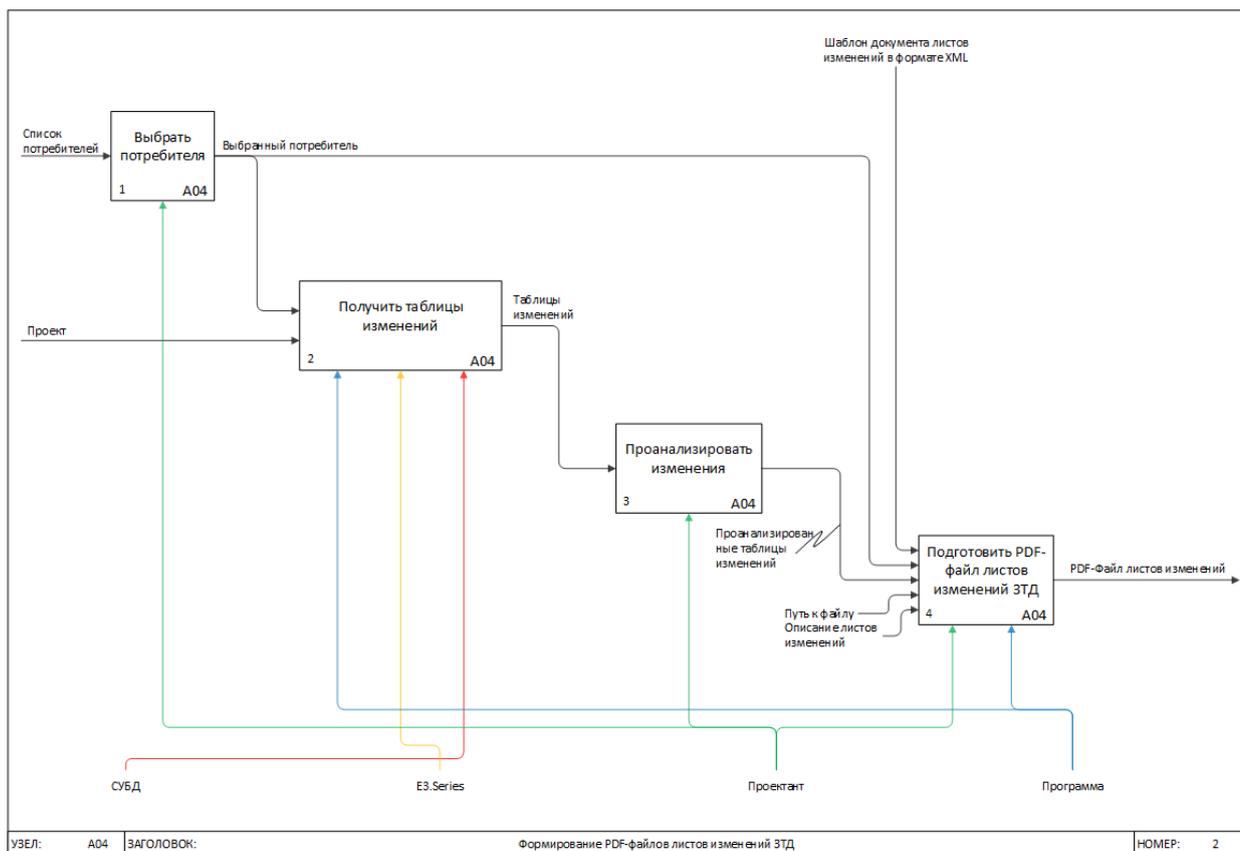


Рисунок 5 – Бизнес-процессы подготовки листов изменений ЗТД к выдаче потребителю

2.4 Анализ инструментов разработки приложений для персонального компьютера

В современном мире имеется множество языков программирования начиная от низкоуровневых, заканчивая высокоуровневыми, в частности для разработки на платформе *Windows* выделены основные популярные, такие как: *C/C++*, *C#*, *Java*, *Python*. В данной работе за основу взят язык программирования *C#*, так как он является быстрым, удобным (по сравнению с *C* и *C++*) и эффективным языком для создания приложений для персонального компьютера, имеет широкий спектр инструментария и сообщество программистов.

2.4.1 Фреймворки для создания приложений для персонального компьютера

Windows Forms

Интерфейс программирования приложений (*API*), отвечающий за графический интерфейс пользователя и являющийся частью *Microsoft .NET Framework*. Данный интерфейс упрощает доступ к элементам интерфейса *Microsoft Windows* за счет создания обёртки для существующего *Win32 API* в управляемом коде. Причём управляемый код – классы, реализующие *API* для *Windows Forms*, не зависят от языка разработки. То есть программист одинаково может использовать *Windows Forms* как при написании ПО на *C#*, *C++*, так и на *VB.Net*, *J#* и др.

Приложение *Windows Forms* представляет собой событийно-ориентированное приложение, поддерживаемое *Microsoft .NET Framework*. В отличие от пакетных программ, большая часть времени тратится на ожидание от пользователя каких-либо действий, как, например, ввод текста в текстовое поле или клика мышкой по кнопке [13].

Это *API* вытеснено более новым и современным фреймворком *Windows Presentation Foundation (WPF)* поэтому на данный момент он считается устаревшим.

Windows Presentation Foundation (WPF)

Аналог *WinForms*, система для построения клиентских приложений *Windows* с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе *.NET Framework* (начиная с версии 3.0), использующая язык *XAML*.

XAML (англ. *eXtensible Application Markup Language*) – расширяемый язык разметки для приложений — основанный на *XML* язык разметки для декларативного программирования приложений, разработанный *Microsoft*.

В основе *WPF* лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. *WPF* предоставляет средства для создания визуального интерфейса, включая язык *XAML* (*eXtensible Application Markup Language*), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление. Графической технологией, лежащей в основе *WPF*, является *DirectX*, в отличие от *Windows Forms*, где используется *GDI/GDI+*. Производительность *WPF* выше, чем у *GDI+* за счёт использования аппаратного ускорения графики через *DirectX*. Также существует урезанная версия *CLR*, называемаяся *WPF/E*, она же известна как *Silverlight* [14].

WPF предоставляет широкий спектр возможностей по созданию интерактивных настольных приложений:

Привязка данных

Это гибкий механизм, который позволяет через расширения разметки *XAML* связывать различные данные (от значений свойств элементов управления до общедоступных свойств, реализующих поля базы данных через *Entity Framework*). Привязка данных представлена классом *Binding*, который в свою очередь унаследован от *MarkupExtension*, что позволяет использовать привязки не только в коде, но и в разметке [14].



Графика

WPF представляет обширный, масштабируемый и гибкий набор графических возможностей:

- Графика, не зависящая от разрешения и устройства. Основной единицей измерения в графической системе *WPF* является аппаратно-независимый пиксель, который составляет 1/96 часть дюйма независимо от фактического разрешения экрана.
- Дополнительная поддержка графики и анимации. *WPF* упрощает программирование графики за счет автоматического управления анимацией. Разработчик не должен заниматься обработкой сцен анимации, циклами отрисовки и билинейной интерполяцией
- Аппаратное ускорение. Графическая система *WPF* использует преимущества графического оборудования, чтобы уменьшить использование ЦП [14].

Двухмерная графика

WPF предоставляет библиотеку общих двухмерных фигур, нарисованных с помощью векторов, таких, как прямоугольники и эллипсы, а также графические пути. И в своей функциональности фигуры реализуют многие возможности, которые доступны обычным элементам управления.

Двухмерная графика в *WPF* включает визуальные эффекты, такие как градиенты, точечные рисунки, чертежи, рисунки с видео, поворот, масштабирование и наклон.

Трехмерная графика

WPF также включает возможности трехмерной отрисовки, интегрированные с двухмерной графикой, что позволяет создавать более яркий и интересный пользовательский интерфейс [14].

2.4.2. Паттерны проектирования

Model-View-Controller (MVC) – фундаментальный паттерн, который нашел применение во многих технологиях, дал развитие новым технологиям и каждый день облегчает жизнь программистам, он описывает простой способ

построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и конечно же реализуется [15].

Описание компонентов:

Модель – это данные приложения, логика их получения и сохранения. Зачастую это модель предметной области, основанная на базе данных или на результатах от веб-сервисов.

View отвечает за отображение *UI* на экране. Без библиотек виджетов, это означало самостоятельную отрисовку блоков, кнопок, полей ввода и тому подобное, *View* также может наблюдать за моделью и отображать данные из неё.

Controller обрабатывает действия пользователя и затем обновляет *Model* или *View*. Если пользователь взаимодействует с приложением (нажимает кнопки на клавиатуре, передвигает курсор мыши), контроллер получает уведомление об этих действиях и решает, что с ними делать [15].

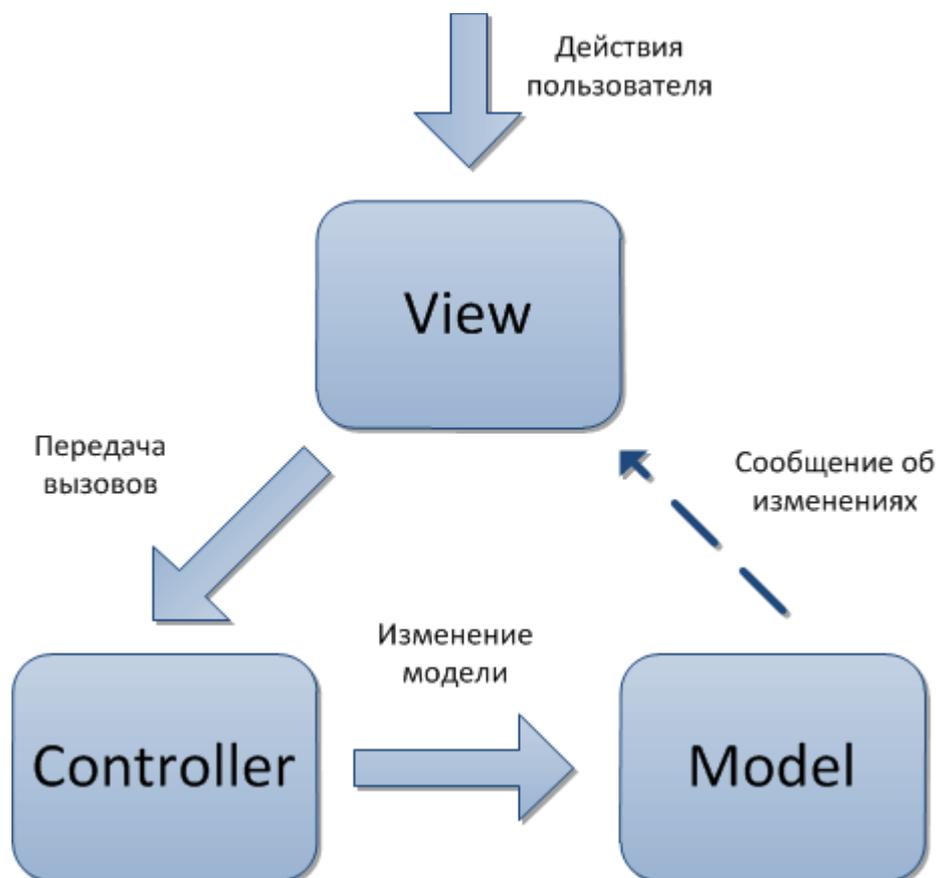


Рисунок 6 – Паттерн *MVC* [15]

Model-View-ViewModel (MVVM) — шаблон проектирования архитектуры приложения, ориентирован на современные платформы разработки, такие как *Windows Presentation Foundation*.

Шаблон MVVM делится на три части:

- Модель (англ. *Model*) (так же, как в классической *MVC*) представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения.
- Представление (англ. *View*) – графический интерфейс (окна, списки, кнопки и т. п.). Выступает подписчиком на событие изменения значений свойств или команд, предоставляемых Моделью Представления. В случае, если в Модели Представления изменилось какое-либо свойство, то она оповещает всех подписчиков об этом, и Представление, в свою очередь, запрашивает обновлённое значение свойства из Модели Представления. В случае, если пользователь воздействует на какой-либо элемент интерфейса, Представление вызывает соответствующую команду, предоставленную Моделью Представления.
- Модель Представления (англ. *ViewModel*) – с одной стороны, абстракция Представления, а с другой – обёртка данных из Модели, подлежащих связыванию. То есть, она содержит Модель, преобразованную к Представлению, а также команды, которыми может пользоваться Представление, чтобы влиять на Модель [15].



Рисунок 7 – Схема паттерна проектирования *MVVM* [15]

Выбран именно этот паттерн проектирования, потому что является основным при работе с фреймворком *WPF*, так как имеет возможность автоматически привязываться к данным модели, без написания излишнего кода с использованием механизма привязки данных «*Binding*». Отказ от *MVC (Model-View-Controller)* был основанным: этот паттерн используется в ситуациях, когда связь между представлением и другими частями приложения – невозможна (зачастую он используется в *ASP.NET MVC*)

2.4.3 Выбор среды разработки

Интегрированная среда разработки (*IDE*) – это многофункциональная программа, которая поддерживает многие аспекты разработки программного обеспечения.

Интегрированная среда разработки *Visual Studio* – это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред *IDE*, *Visual Studio* включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки [17].

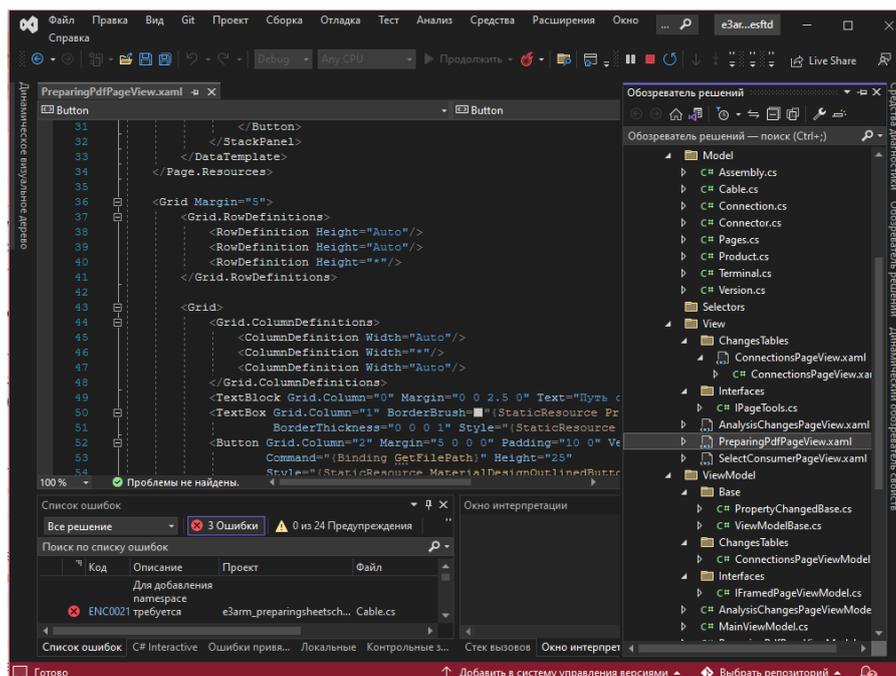


Рисунок 8 – Среда *Visual Studio*

Плюсы:

- Официальная. Так как и язык, и среда разработки созданы в *Microsoft*, логично предположить, что ничего более функционального вы не найдете во всем Интернете. В некоторых случаях без *Visual Studio* не обойтись – например, при использовании технологий *UWP* и *WPF*.
- Бесплатная. Версии «*Community edition*» для рядового пользователя будет достаточно. Тем более, теперь можно подключать плагины (в отличие от старой версии *Express*).
- Функциональная. В *Visual Studio* множество качественных плагинов. С их помощью можно расширить функциональность приложения и подключить другие языки.
- Поддерживает платформы *.NET*. *Visual Studio* имеет широкие возможности по разработке приложений под *Windows*, в том числе в *.NET*-сегменте.
- Облачные хранилища. Зарегистрируйтесь в сообществе *Visual Studio* – и получите доступ к облачному хранилищу, где сможете располагать файлы проектов.
- Корпоративность. Технология бэклога позволяет членам команды взаимодействовать при гибкой методологии разработки.

Минусы:

- Баги при переходах с триал-версии. При переходе на платную версию могут теряться настройки и нарушаться работа корпоративного сервера.
- Сложность. Самостоятельно освоить *Visual Studio* новичку будет непросто – слишком много доступных функций, спрятанных в подразделах меню [17].

Project Rider (JetBrains) – новая среда разработки, ещё не зарекомендовавшая себя на рынке [19]

Плюсы:

- *ReSharper*. Это плагин, изначально разработанный для повышения производительности *Visual Studio*. Теперь на его основе выпущена *IDE*.
- Поддержка полного цикла. Фирменная черта продуктов *JetBrains*, воплощенная и в *Project Rider*. С ним вы сможете организовать весь цикл создания ПО: от идеи до поддержки.

- Функциональность. *Project Rider* позволяет подключить *MSBuild* и *XBuild*, работать с CLI-проектами и организовать отладку приложений *.NET and Mono*. Множество опций для быстрого создания кода улучшает производительность.
- Кроссплатформенность. *Project Rider* работает с *Windows, Linux* и *MacOS*.
- Контроль версий. Встроенный инструмент позволяет напрямую организовать работу с *Git, Mercurial* и *TFS*.

Минусы:

- Молодость. Часть функциональности еще в разработке, не все стартовые баги исправлены.
- Стоимость. Самая дешевая версия *Project Rider* обойдется в 139 долларов за первый год использования. Но есть триал-версия и специальные предложения для студентов и непрофильных организаций [19].

Исходя из представленных *IDE* была выбрана *Visual Studio*, несмотря на минусы она является популярнейшим и, к тому же, бесплатным (в версии *Community*), удобным и функциональным приложением для разработки не только под систему *Windows*.

2.4.4 Описание инструмента доступа и редактирования СУБД организации

SQL Server Management Studio (SSMS) – это бесплатная графическая среда, включающая набор инструментов для разработки сценариев на T-SQL и управления инфраструктурой Microsoft SQL Server.

Среда SQL Server Management Studio – это основной, стандартный и полнофункциональный инструмент для работы с Microsoft SQL Server, разработанный компанией Microsoft, который предназначен как для разработчиков, так и для администраторов SQL Server [18].

Также в задачи входит полное сопровождение Microsoft SQL Server, начиная от создания баз данных, написания SQL запросов, создания хранимых процедур и функций, и заканчивая администрированием SQL Server, включая управление безопасностью.

Обозреватель объектов среды SQL Server Management Studio – это графический пользовательский интерфейс для просмотра и управления объектами в каждом экземпляре SQL Server; предоставляет интерфейс, в котором структура объектов сервера представлена в очень удобном иерархическом виде, напоминает что-то вроде файловой системы Windows, т.е. папки, подпапки, файлы и т. д. [18].

Шаблоны в SQL Server Management Studio – это файлы, содержащие стандартный SQL код, который предназначен для создания объектов на SQL Server – это готовые SQL скрипты, содержащие заголовки SQL инструкций, в которые необходимо подставить данные, например, имена объектов, чтобы создать тот или иной объект на SQL Server [18].

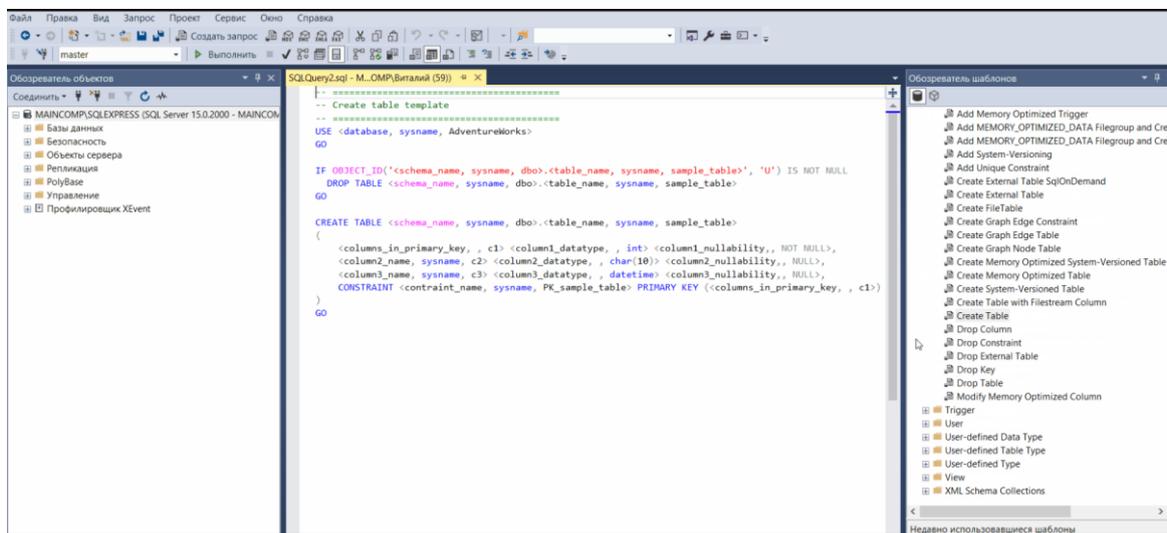


Рисунок 9 – Microsoft SQL Server Management Studio

Выводы

Предметной областью является документация на продукцию предприятия. Потребителями данной документации являются производство предприятия и заказчик, а также проектные институты. Данные потребители на протяжении всего жизненного цикла разработки нуждаются в листах изменений, для более быстрого и понятного изучения и сравнения двух версий ЗТД. На основании анализа предметной области были составлены соответствующие модели бизнес-процессов.

Для разработки продукта был выбран язык программирования C# как самый эффективный в сфере разработки приложений для персонального компьютера с использованием операционной системы Windows. Фреймворк WPF вы-

бран потому что, является более гибким и быстрым за счет использования DirectX по сравнению Windows Forms. Также паттерн проектирования MVVM был выбран потому что, позволяет в структуре приложения позволяет разделять интерфейс и данные, что присуще WPF.

Проделанная работа позволит эффективно и качественно продолжить работу над разработкой приложения: построить логику, спроектировать базу данных и интерфейс приложения.

3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Важным этапом в разработке программного обеспечения является прототипирование. Этот этап может содержать под этапы такие как: описание основной логики и логики отдельных функций, описание и создание прототипа визуальной части приложения, построение модели базы данных, модели классов и многое другое. В данной главе будут рассмотрены все перечисленные принципы создания прототипа интерфейса.

3.1 Построение логики приложения

В прошлой главе были проанализированы бизнес-процессы подготовки заводской технической документации к выдаче потребителю, так же на основе анализа были составлены соответствующие схемы-модели бизнес-процессов. Данная проделанная работа дала более глубокое понимание предметной области, что повлечет создание качественного продукта. На основе полученной информации можно сделать обоснованный вывод, что процесс подготовки листов изменений заводской технической документации можно разделить на три основных этапа:

- Этап выбора потребителя;
- Этап анализа полученных изменений;
- Этап подготовки *PDF*-файлов заводской технической документации.

Исходя из результатов анализа технического задания, представленного заказчиком продукта, можно сделать следующее описание данных этапов:

Этап выбора потребителя – проектировщику, ответственному за подготовку заводской технической документации к выдаче потребителю выдается задание с информацией о потребителе, которому нужно выдать документацию.

Данный программный продукт должен позволять:

- выбирать из справочника потребителя продукции, набор которых является статическим для всех других проектов;
- ответственному лицу просматривать информацию о других фактах передачи проекта потребителям в соответствующей таблице:

Таблица 1 – Макет таблицы фактов передачи документации потребителю

№	Ф. И. О. Ответственного	Комментарий
---	----------------------------	-------------

- Отменять и восстанавливать факты передачи документации;
- Просматривать историю удаления/восстановления фактов передачи документации:

Таблица 2 – Макет таблицы истории факта передачи документации

Действие	Дата	Ф. И. О. Ответственного	Комментарий
----------	------	----------------------------	-------------

Итогом данного этапа должны быть предварительные таблицы изменений, которые проектировщик должен будет проанализировать и выявить ошибки и недочеты.

Этап анализа полученных изменений – итогом предыдущего этапа являются предварительные таблицы изменений по каждому отслеживаемому объекту, в данной работе рассматриваются объекты соединений и таблицы изменений к ним. Данный этап является важным в процессе подготовки, так как исключается человеческий фактор – ошибки при проектировании, мелкие недочеты; позволяет повысить качество получаемой документации. Таблица, представляющая изменения объектов соединений представлена в Приложении А1.

Итогом данного этапа выступают проанализированные таблицы изменений, готовые к выгрузке в формат файла *PDF*.

Этап подготовки *PDF*-файлов заводской технической документации – завершающий этап подготовки документации к выдаче. На данном этапе решаются задачи по выбору места сохранения получаемых файлов, написание подробного описания к версии документации.

На основе данного глубокого анализа этапов подготовки документации можно составить следующую схему-модель «Логика приложения» на основе нотации *VRMN*, рисунок 10.

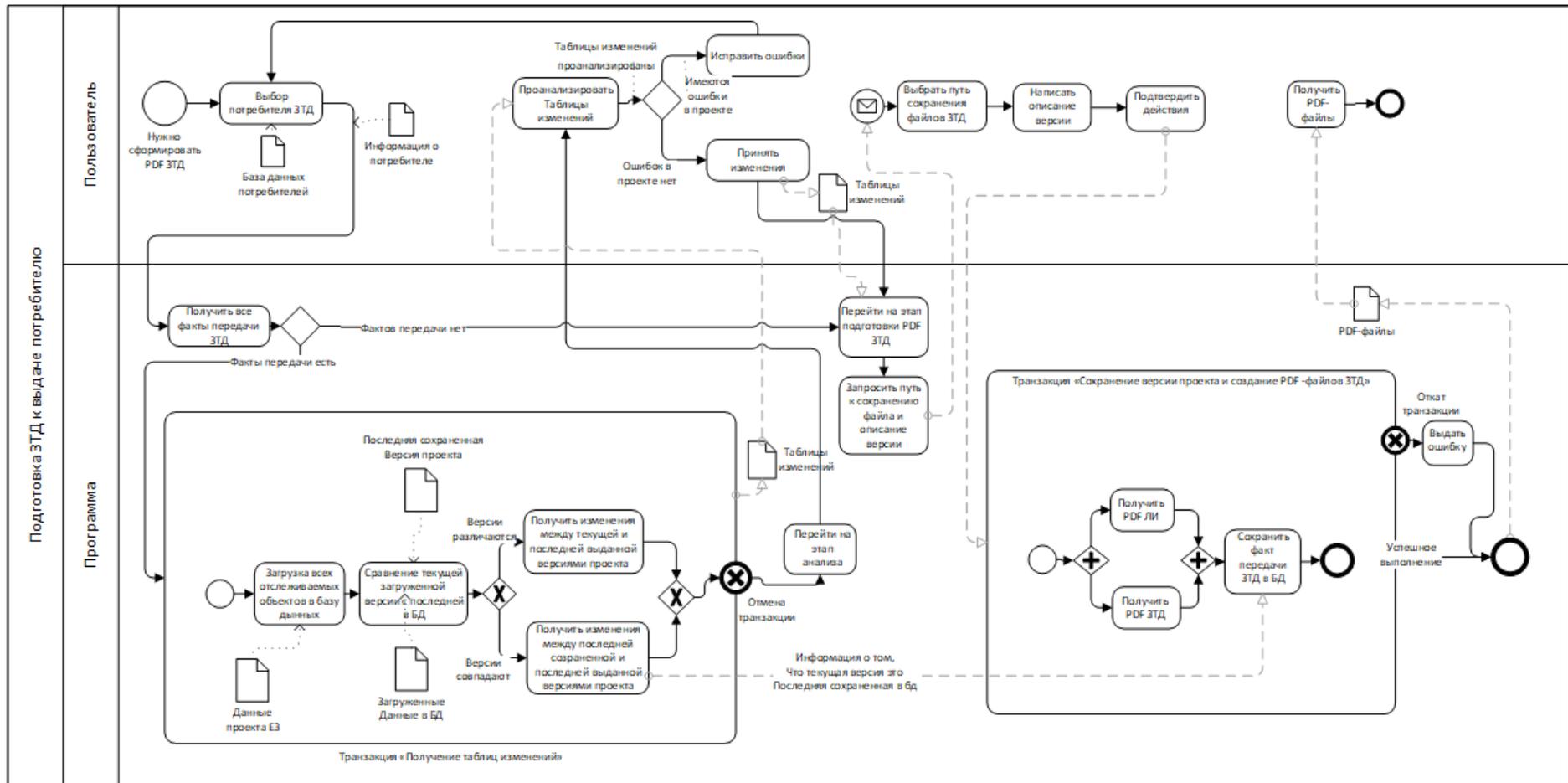


Рисунок 10 – Схема-модель логика работы приложения в виде нотации *BPMN*

3.2 Проектирование структуры базы данных

В соответствии с техническим заданием можно выделить отслеживаемые объекты проекта *E3.Series*:

- Соединения и их адресация.

Соединения (относительно проекта в САПР) – объект, соединяющий 2 и только 2 изделия, например, изделие1: пин1 – изделие2: пин2. В соответствии с требованиями компоновки они не должны иметь свободные концы или 3 и более концов подключения. Приложение А1 к настоящей ВКР описывает пример результата работы программы – таблица изменений соединений; в результате анализа можно выделить следующие атрибуты соединений для хранения в базе данных:

- Откуда устройство (шкаф – понятие в рамках предприятия);
- Откуда место (отсек шкафа – понятие в рамках предприятия);
- Откуда позиционное обозначение;
- Откуда пин изделия;
- Куда устройство (шкаф – понятие в рамках предприятия);
- Куда место (отсек шкафа – понятие в рамках предприятия);
- Куда позиционное обозначение;
- Куда пин изделия;
- Имя цепи соединения;
- Кабель/провод, используемый для реализации соединения.

В будущей базе данных, в соответствии с требованиями технического задания, нужно хранить также информацию:

- Потребители;
- Проекты САПР, для которых создаются факты передачи;
- Версии проектов с привязкой к отслеживаемым объектам;
- Историю изменений фиксации фактов передачи;
- Факты передачи потребителю.

Исходя из анализа можно выделить следующие сущности и их атрибуты:

Проекты:

- Имя;
- Имя компьютера, где проект используется;
- Путь к файлам проекта;

- Отметка о том, является ли проект многопользовательским.

Версии:

- Дата сохранения;
- Ответственный;
- Описание.

Потребители:

- Имена потребителей.

Факты передачи заводской технической документации:

- Потребитель;
- Версия;
- Актуальность факта передачи.

История изменений фактов передачи:

- Действие;
- Дата действия;
- Ответственный;
- Описание действия;
- Отметка о том, сделано ли действие системой.
- Соединения;
- Изделие откуда;
- Пин откуда;
- Изделие куда;
- Пин куда;
- Имя цепи;
- Кабель/провод.

Изделия:

- Уникальный идентификатор;
- Сборка, если изделие является составляющей сборки;
- Позиционное обозначение;
- Устройство;
- Место;
- Тип изделия (Клемма, разъем, девайс);
- Отметка о том, является ли изделие сборкой;
- Прибор.

Кабели:

- Уникальный идентификатор;
- Устройство откуда;
- Устройство куда;
- Позиционное обозначение;
- Длина кабеля;
- Номенклатура кабеля.

Жилы кабелей:

- Уникальный идентификатор;
- Имя;
- Тип;
- Кабель, в котором находится жила.

Используя построенные сущности была реализована модель базы данных, рисунок 11.

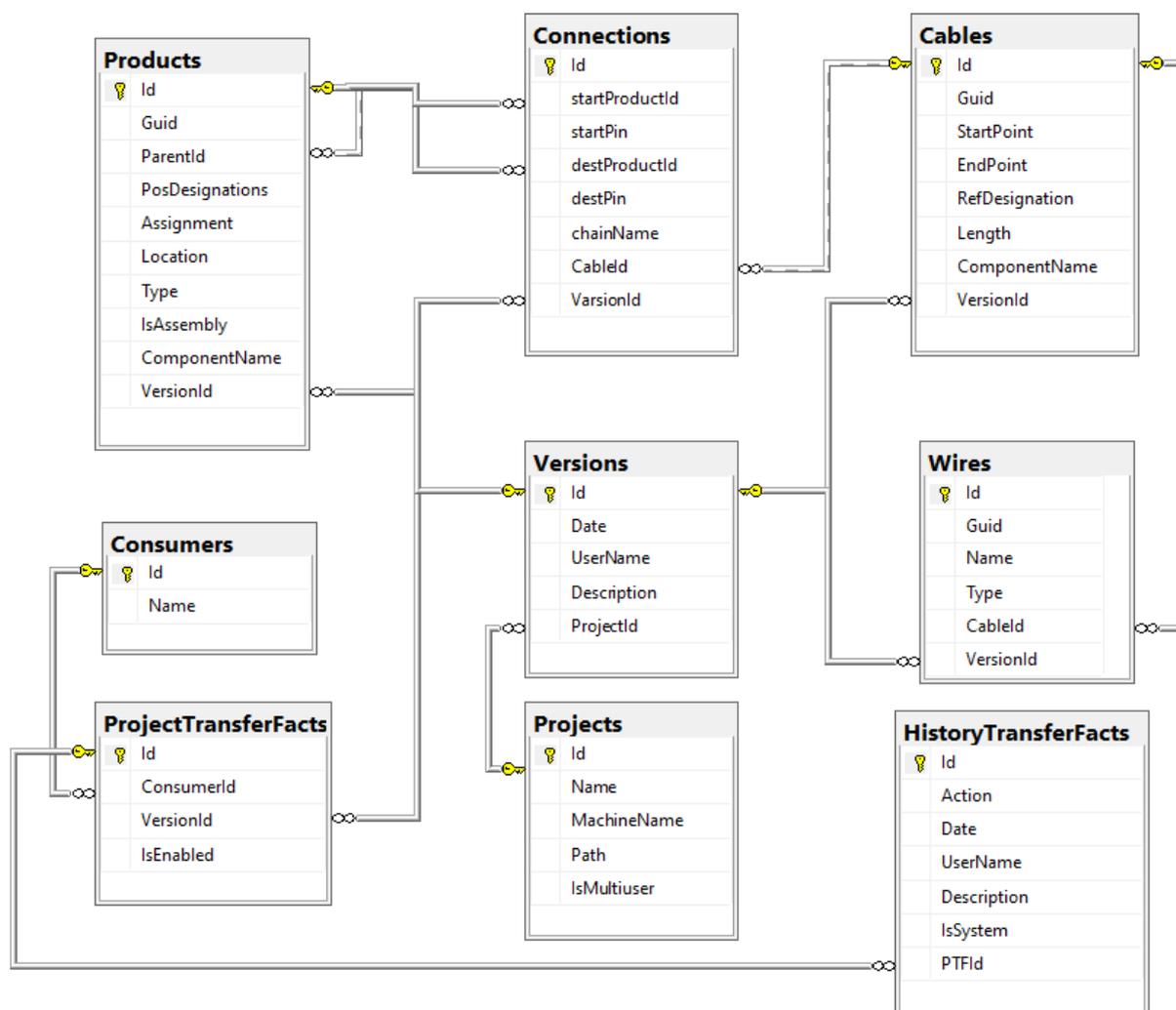


Рисунок 11 – Дatalogическая модель базы данных

3.3 Прототипирование интерфейса приложения

Прототипирование интерфейса было выполнено с использованием языка разметки *XAML*, интегрированного в описанный ранее фреймворк *WPF*. В соответствии с техническим заданием подобрана цветовая гамма и расположение элементов интерфейса, все наименования элементов выполнены с соблюдением логики работы приложения. Каждая страница интерфейса выполнена в одном стиле: шапка страницы содержит заголовок страницы (этапа), также панель инструментов; размер шрифта 14 кеглей.

Исходя из предложенной ранее логики работы приложения первым этапом был выделен «Выбор потребителя ЗТД» см. Рисунок 3, а также основные элементы страницы интерфейса:

- Список потребителей с возможностью выбора;
- Кнопка перехода на следующий этап «Посмотреть изменения»;
- Таблица, отражающая информацию о фактах передачи ЗТД конкретному потребителю, выбранному в списке потребителей. Таблица имеет 2 типа записей:
 - Факт передачи – отражает информацию о дате, ответственном и описании версии проекта, обозначается белым цветом. Должна поддерживаться функция отмены факта передачи, выполненная в контекстном меню см. Рисунок 4;
 - Отмененный факт передачи – отражает информацию о дате, ответственном и описании версии проекта, обозначается серым цветом. Должна поддерживаться функция восстановления факта передачи, выполненная в контекстном меню см. Рисунок 5.

Также у каждого элемента таблицы (факта передачи) должна быть история изменений, переход на страницу, которой расположен в контекстном меню см. Рисунок 6. На странице «Просмотр изменений факта передачи ЗТД» должна быть таблица, содержащая информацию об изменении факта передачи с течением времени см. рисунок 12, 13, 14.

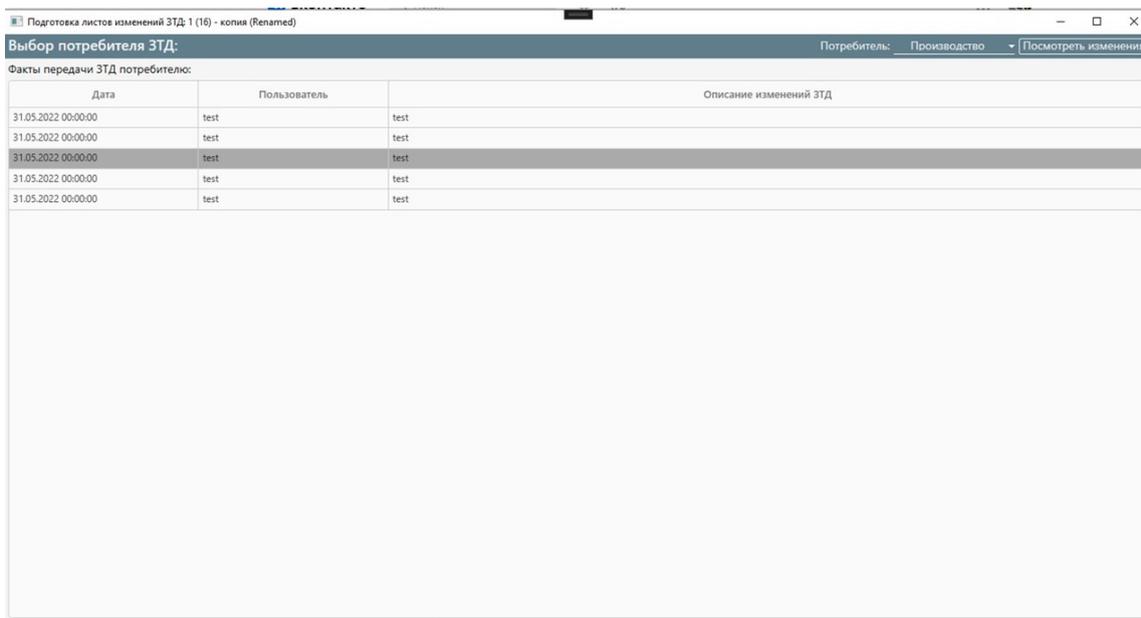


Рисунок 12 – Интерфейс «Выбор потребителя ЗТД»

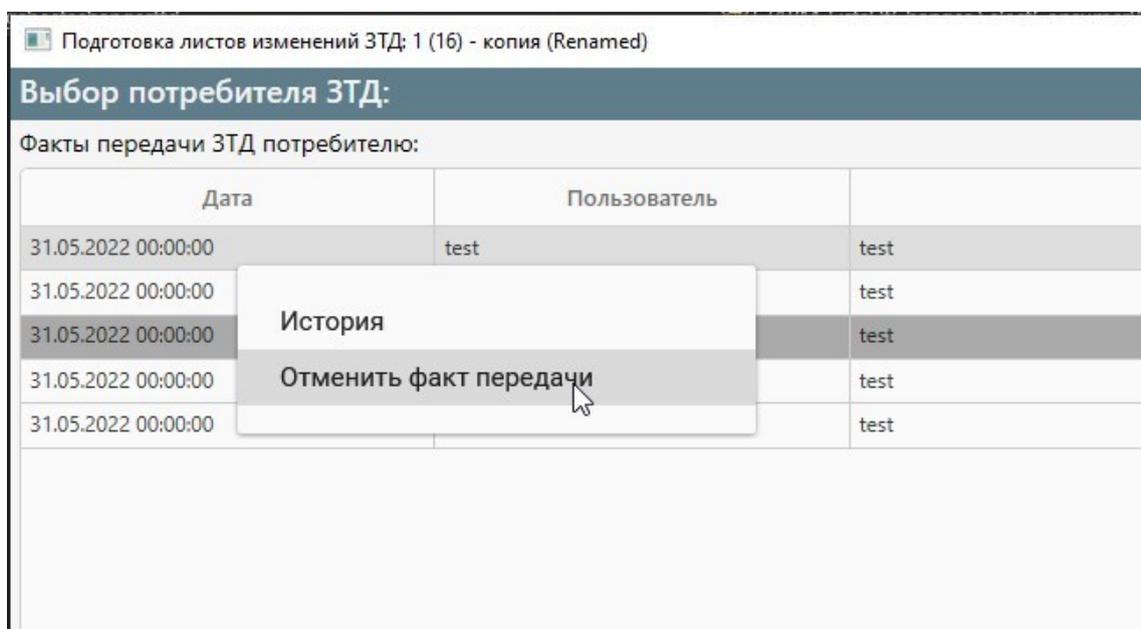


Рисунок 13 – Контекстное меню. Пункт «Отмена факта передачи ЗТД»

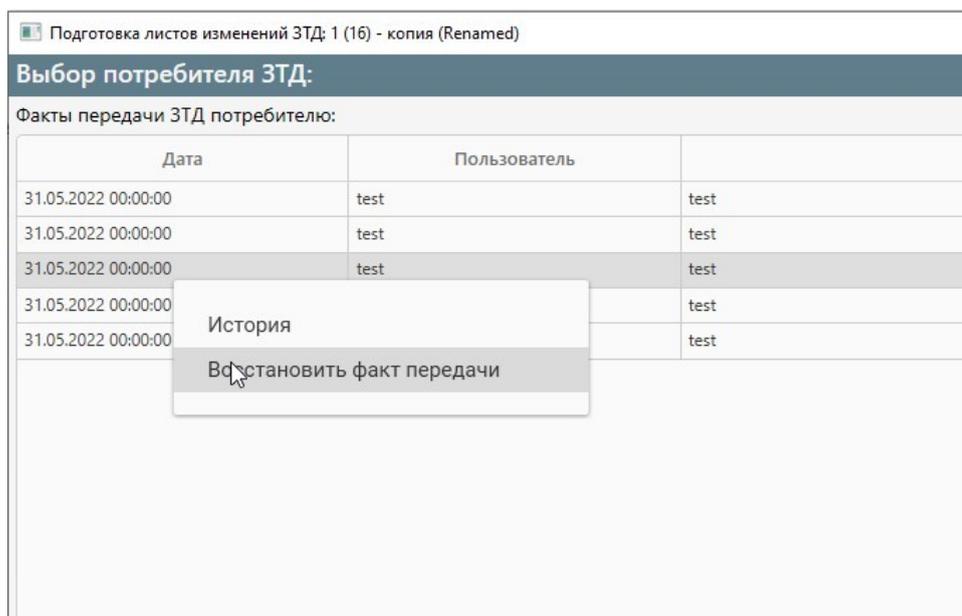


Рисунок 14 – Контекстное меню. Пункт «Восстановить факт передачи»



Последний этап – Подготовка *PDF*-файлов листов изменений ЗТД. На данном этапе пользователь выбирает путь к сохранению и задает краткое описание изменений. Тем самым интерфейс должен поддерживать ввод текста и поле выбора пути сохранения файла, пример представлен на рисунке 9.

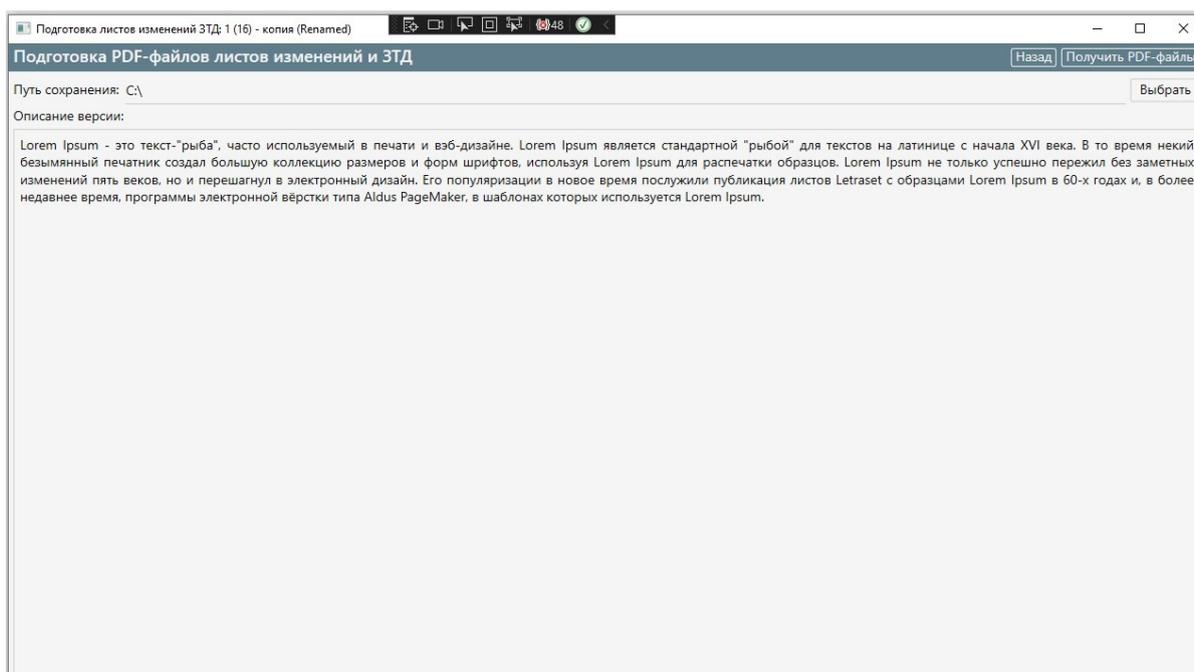


Рисунок 9 – Интерфейс «Подготовка *PDF*-файлов листов изменений ЗТД»

Подводя итог прототипирования интерфейса можно сделать следующие выводы: спроектированный прототип интерфейса удовлетворяет интересам заказчика и техническому заданию, удобен и понятен для начинающего пользователя приложения, что является важным итогом работы, так как не нужно выделять дополнительное время для его изучения.

3.4 Создание базы данных

На основании схемы, разработанной в главе 2 настоящей ВКР, будут созданы две базы данных на сервере *MS SQL Server* организации. Первая база будет использоваться пользователями и считаться основной рабочей, вторая – для разработки и тестирования, будет использоваться разработчиком продукта. Принципы создания этих баз не отличаются поэтому будет писана методика создания основной рабочей базы данных, способы подключения к ней, запросы, которые будут использоваться в коде программы.

3.4.1 Создание базы данных на сервере *MS SQL Server*

Для создания базы данных нужен доступ к серверу, который предоставляется системным администратором организации. Под доступом понимается логин и пароль для входа и конкретные права, определяющие перечень разрешенных действий. Для подключения к рабочему ядру СУБД используется описанный в главе 2 инструмент *MS SQL Server Management Studio*, имеющий похожий интерфейс с *MS Visual Studio* (рисунок 15).

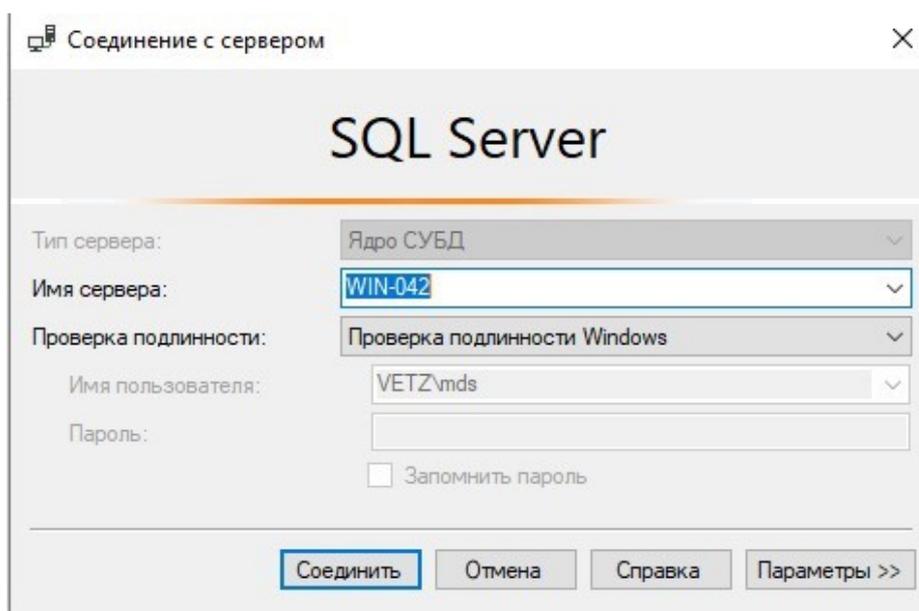


Рисунок 15 – Окно подключения к ядру СУБД с полученными данными доступа

На панели «Обозреватель решений» выделены разделы с существующими БД на сервере в родительском разделе «Базы данных». Для того, чтобы её создать нужно нажать на правую кнопку мыши по родительскому разделу для вызова контекстного меню, описанного на рисунке 16 настоящей главы и выбрать пункт «Создать базу данных».

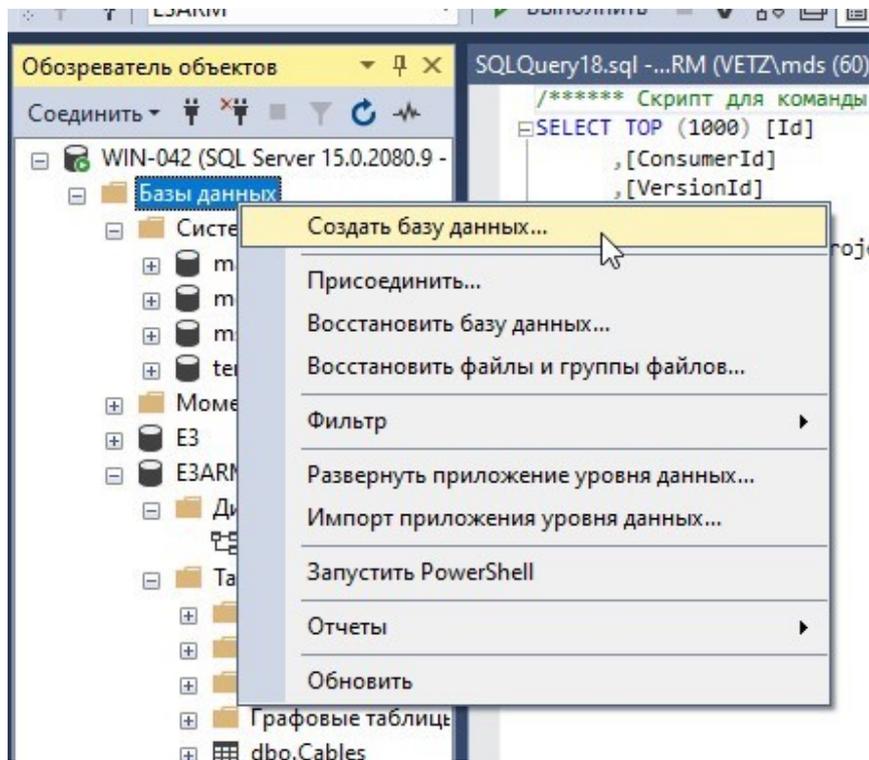


Рисунок 16 – Контекстное меню родительского раздела «Базы данных»

В появившемся окне (см. Рисунок 17) есть возможность выбрать имя, владельца, путь к файлам, способ сортировки – как основные базовые параметры настройки, создаваемой БД.

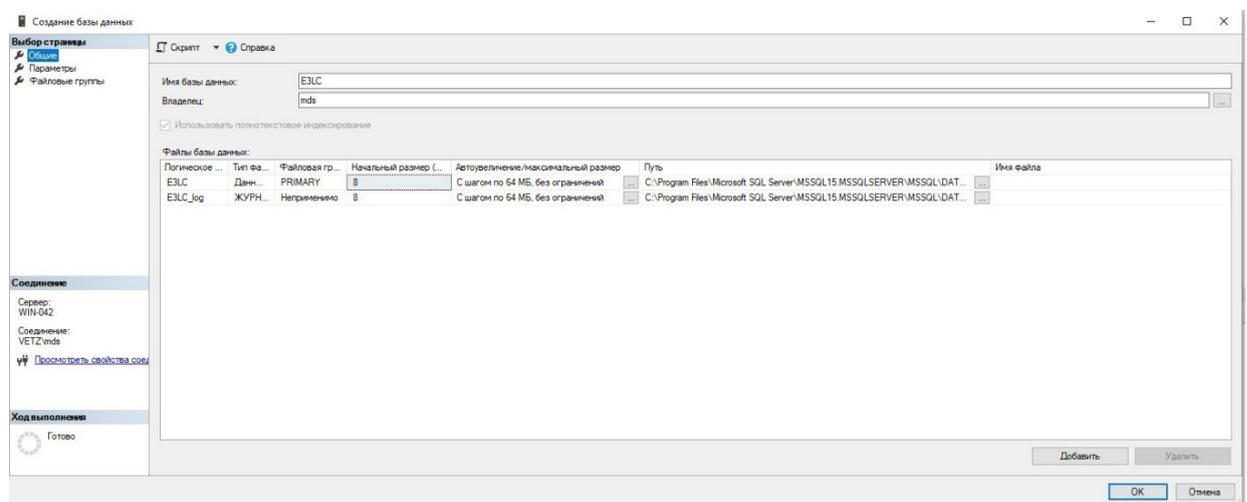


Рисунок 17 – Окно настроек создаваемой базы данных

В результате данного действия владельцем базы данных был выбран разработчик, также был создан пользователь, имеющий права доступа «только для чтения», логин и пароль которого будут использоваться для подключения к настоящей базе из кода программы.

3.4.2 Создание диаграммы БД средствами *MS SQL Server Management Studio*

Существуют способы создания сущностей БД:

- Запросом на языке *SQL*;
- С помощью встроенных средств;
- Автоматически с использованием *ER*-диаграммы БД.

В настоящей ВКР будет использоваться способ создания сущностей с использованием *ER*-диаграммы, которая была смоделирована в главе 2.

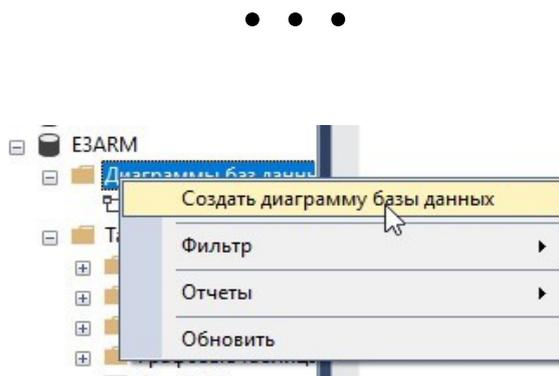


Рисунок 18 – Контекстное меню «Диаграммы баз данных»

Добавление сущностей осуществляется нажатием правой кнопкой мыши по рабочей области инструмента и выбором в контекстном меню пункта «Создать таблицу».

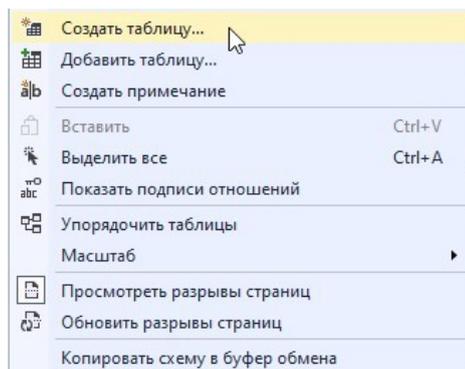


Рисунок 19 – Контекстное меню рабочей области

После выбора имени сущности описывается каждый атрибут и осуществляется выбор типа данных. Связи между сущностями добавляются с помощью инструмента «Создание связей», который имеет полный набор функций по созданию связей по атрибутам двух сущностей и ограничений ключей.

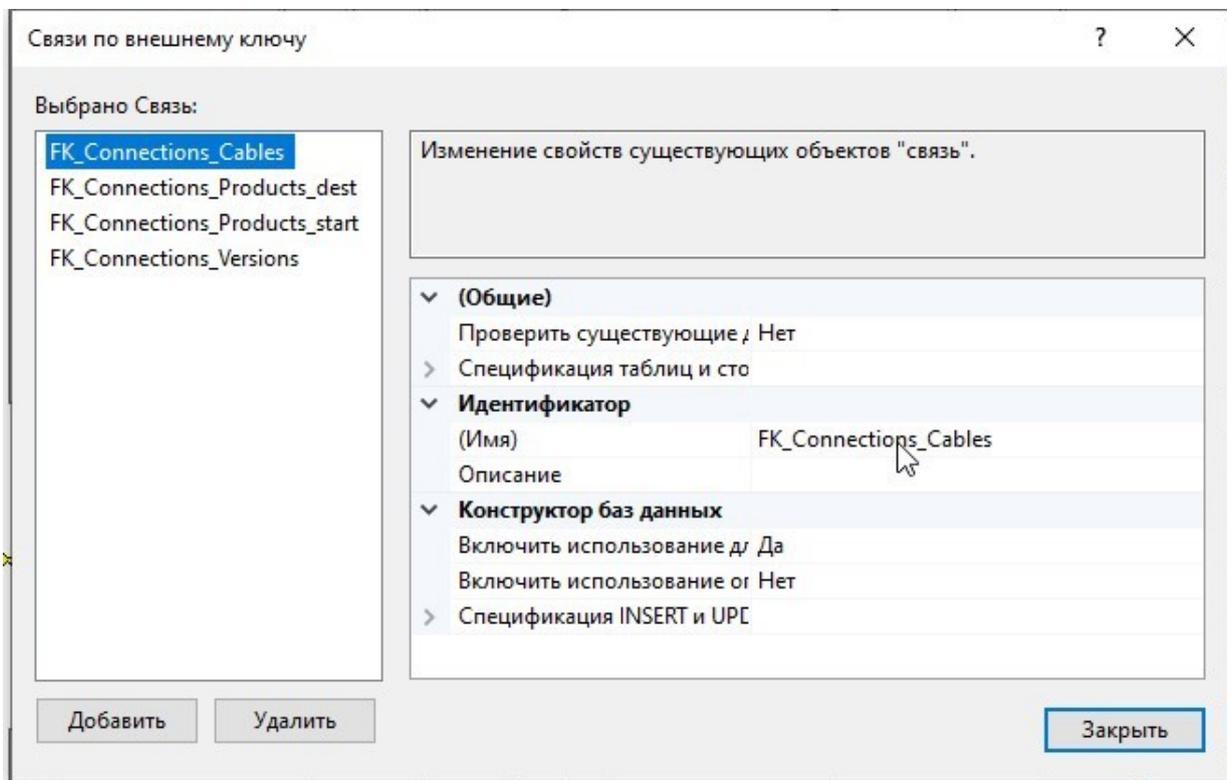


Рисунок 20 – Инструмент «Создание связей»

В результате проделанной работы разработана модель, которая была описана в главе 3 настоящей ВКР, *MS SQL Server Management Studio* на её основании в автоматическом режиме создала соответствующие таблицы.

3.5 Разработка интерфейсной части приложения

В соответствии с прототипами, описанными в главе 2, были созданы следующие страницы на языке разметки *XAML*:

- *MainWindow* – основное окно, содержащее страницы;
- *SelectConsumerPageView* – страница выбора потребителя;
- *ProjectTransferStoryPageView* – страница просмотра истории изменений факта передачи ЗТД потребителю;
- *AnalysisChangesPageView* – страница просмотра изменений проекта;
- *ConnectionsPageView* – подстраница, которая относится к *AnalysisChang-*

esPageView, содержащая в себе таблицу изменений соединений;

– *PreparingPdfPageView* – страница подготовки *PDF*-файлов листов изменений ЗТД с полями описание изменений и выбор пути.

Основное окно приложения содержит основную информацию и панель инструментов для каждой страницы, а также объект управления перемещением между страницами *Frame*, который уже содержит в себе страницы. Панель в основном окне спроектирована в соответствии с прототипами:



Рисунок 21 – Панель главного окна

```
<Grid Background="{StaticResource WindowToolBarColor}">
  <TextBlock Margin="5 0" Text="{Binding PageSource.Title}"
    Style="{StaticResource TextStyle}"
    Foreground="{StaticResource WindowToolBarTextColor}"
    HorizontalAlignment="Left" FontSize="18" FontWeight="SemiBold"/>
  <ContentControl HorizontalAlignment="Right" x:Name="toolBar"
    ContentTemplate="{Binding PageSource,
      Converter={StaticResource PageSourceToTemplateConverter},
      Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"/>
</Grid>
```

Рисунок 22 – Код на языке разметки *XAML* панели главного окна

В данном случае объект *TextBlock* представляет собой заголовок текущей открытой страницы, его атрибут *Binding* определяет контекст данных, который будет привязан во время выполнения и соответствует паттерну проектирования *MVVM*, как разделение визуальной части приложения и данных. Объект *ContentControl* определяет контент панели инструментов для каждой страницы, который привязан к свойству *PageSource* основной *ViewModel*, с помощью этого главное окно будет знать о существующих страницах.

Страница «Выбор потребителя ЗТД» содержит таблицу о фактах передачи с возможностью просмотра их изменений. На рисунке 24 представлен листинг кода на языке разметки *XAML* реализующий данную таблицу.

Дата	Пользователь	Описание изменений ЗТД
31.05.2022 00:00:00	test	test
06.06.2022 00:00:00	test	test
06.06.2022 00:00:00	test	test

Рисунок 24 – Таблица фактов передачи ЗТД потребителю

```

<DataGrid x:Name="projectTransferFactsDataGrid"
  Margin="0 2.5 0 0" Grid.ColumnSpan="2"
  Grid.Row="1" Grid.Column="0" ItemsSource="{Binding ProjectTransferFacts}"
  BorderBrush="{StaticResource SelectConsumerPageDataGridBorder}"
  BorderThickness="1"
  CanUserAddRows="False" CanUserDeleteRows="False"
  CanUserResizeRows="False" CanUserReorderColumns="False"
  GridLinesVisibility="All"
  SelectionUnit="FullRow" IsReadOnly="True"
  AutoGenerateColumns="False"
  VirtualizingPanel.IsVirtualizing="True"
  VirtualizingPanel.VirtualizationMode="Recycling">
  <DataGrid.Resources>...</DataGrid.Resources>
  <DataGrid.CellStyle>...</DataGrid.CellStyle>
  <DataGrid.RowStyle>
    <Style TargetType="DataGridRow" BasedOn="{StaticResource MaterialDesignDataGridRow}">
      <Setter Property="Background" Value="{Binding IsEnabled, Converter={StaticResource ProjectTransferFactStateConverter}}"/>
      <Setter Property="ContextMenu">
        <Setter.Value>
          <ContextMenu Width="300">
            <MenuItem Header="История" Command="{Binding HistoryFactCommand}" CommandParameter="{Binding}"/>
            <MenuItem Header="Восстановить факт передачи" Command="{Binding RecoveryFactCommand}" CommandParameter="{Binding}"/>
            <MenuItem Header="Отменить факт передачи" Command="{Binding DeleteFactCommand}" CommandParameter="{Binding}"/>
          </ContextMenu>
        </Setter.Value>
      </Setter>
      <Style.Triggers>...</Style.Triggers>
    </Style>
  </DataGrid.RowStyle>
  <DataGrid.Columns>
    <DataGridTextColumn Width="*" Header="Дата" ElementStyle="{StaticResource ContentStyle}"/>
    <DataGridTextColumn Width="*" Header="Пользователь" ElementStyle="{StaticResource ContentStyle}"/>
    <DataGridTextColumn Width="4*" Header="Описание изменений ЗТД" ElementStyle="{StaticResource ContentStyle}"/>
  </DataGrid.Columns>
</DataGrid>

```

Рисунок 25 – Листинг программного кода на языке разметки XAML, представляющий таблицу фактов передачи ЗТД потребителю

В данном случае объект *DataGrid* описывает саму таблицу, её колонки и стиль отображения, его свойство *Binding* позволяет привязать контекст данных таблицы во время выполнения. Свойство *RowStyle* содержит разметку контекстного меню, представленного на рисунке 21.

Следующим этапом выступает анализ полученных данных «Просмотр изменений». Заголовок страницы привязывается к полю, описанному выше, а его панель инструментов содержит две кнопки:

- Назад, которая возвращает пользователя к этапу выбора потребителя;
- Перейти к подготовке PDF, позволяет пользователю перейти к следующему этапу подготовки листов изменений.

```

<DataTemplate x:Key="analysisChangesPageTemplate">
  <StackPanel Margin="5" Orientation="Horizontal" HorizontalAlignment="Right">
    <Button Margin="2.5 0" Padding="5 0" VerticalAlignment="Center" Height="20"
      DataContext="{Binding AnalysisChanges, Source={StaticResource ViewModelRepository}}"
      Command="{Binding GoToSelectConsumerPageCommand}"
      Style="{StaticResource MaterialDesignOutlinedButton}"
      BorderBrush="{StaticResource WindowToolBarTextColor}">
      <TextBlock Text="Назад" Style="{StaticResource TextStyle}" FontSize="14"
        Foreground="{StaticResource WindowToolBarTextColor}"/>
    </Button>
    <Button Margin="2.5 0 0 0" Padding="5 0" VerticalAlignment="Center" Height="20"
      DataContext="{Binding AnalysisChanges, Source={StaticResource ViewModelRepository}}"
      Command="{Binding GoToGetPdfPageCommand}"
      Style="{StaticResource MaterialDesignOutlinedButton}"
      BorderBrush="{StaticResource WindowToolBarTextColor}">
      <TextBlock Text="Перейти к подготовке PDF" Style="{StaticResource TextStyle}" FontSize="14"
        Foreground="{StaticResource WindowToolBarTextColor}"/>
    </Button>
  </StackPanel>
</DataTemplate>

```

Рисунок 26 – Кнопка «Назад» и «Перейти к подготовке PDF»

Кнопками в данном случае выступают объекты *Button* обработкой нажатия которых занимается объект, привязанный к атрибуту *Command*. Для того, чтобы главное окно понимало, какую панель инструментов использовать в том или ином случае, данные кнопки находятся в объекте *DataTemplate*, с помощью которого идет привязка к объекту главного окна *ControlTemplate*.



3.6 Разработка программной части приложения

В соответствии с приложением Б, где описана структура классов приложения, а также разработанным интерфейсом с привязками можно выделить следующие классы *ViewModel*:

- *ViewModelBase*, содержащий в себе поле *Title*; наследуется от класса *PropertyChangedBase*;
- *PropertyChangedBase* – вспомогательный класс, который отвечает за динамическое обновление и отработку событий изменения свойств *ViewModel* программы, работает за счет фреймворка *Fody*;
- *MainViewModel* – отвечает за хранение промежуточных данных главного окна приложения;
- *SelectConsumerPageViewModel* – отвечает за хранение промежуточных данных и работу с интерфейсом «Выбор потребителя ЗТД»;
- *AnalysisChangesPageViewModel* – отвечает за хранение промежуточных данных и работу с интерфейсом «Просмотр изменений»;
- *PreparingPdfPageViewModel* – отвечает за хранение промежуточных данных и работу с интерфейсом «Подготовка Pdf-файлов».

Также классы *Model*:

- *Assembly* – представляет сборку;
- *Cable* – представляет отдельный кабель;
- *Connection* – представляет отдельное соединение (не реализуемо в БД без

- объекта изделия);
- Connector – представляет отдельный разъем;
- Consumer – представляет описание потребителя;
- DBWorker – содержит методы взаимодействия с базой данных;
- DBQuery – содержит константные поля, содержащие SQL-запросы к БД;
- EZWorker – содержит методы для взаимодействия EZ с приложением;
- History – представляет отдельную запись истории изменения факта передачи ЗТД потребителю;
- Pages – содержит наименования всех отслеживаемых объектов;
- Product – представляет отдельное изделие;
- ProjectTransferFact – представляет отдельный факт передачи ЗТД потребителю;
- Terminal – представляет отдельную клемму;
- Version – представляет описание версии проекта.

Важным моментом в работе приложения это хранение всех экземпляров классов интерфейса и ViewModel приложения в одной «Репозитории», у каждого класса может быть отличный жизненный цикл:

- Жизнь в рамках жизненного цикла приложения, т. е. имеющий 1 единственный экземпляр;
- Жизнь в рамках 1 обращения к экземпляру, т. е. – временный объект.

3.6.1 Роль Dependency Injection в создаваемом приложении

В «Приложении В» представлены классы-репозитории содержащие выше-описанные объекты. Каждый класс-репозиторий имеет статический метод Init(), отвечающий за инициализацию объектов. В нем ServiceCollection позволяет хранить создаваемые объекты, а ServiceProvider, обращаться к ним. Данные классы относятся к пространству имен Microsoft.Extensions.DependencyInjection.

Dependency injection (DI) или внедрение зависимостей представляет механизм, который позволяет сделать взаимодействующие в приложении объекты слабосвязанными. Такие объекты связаны между собой через абстракции, например, через интерфейсы, что делает всю систему более гибкой, более адап-

тируемой и расширяемой. [4] Данный паттерн позволяет внедрить в конструктор каждого класса любой написанный сервис, добавленный в ServiceCollection.

ServiceProvider – реализация стандартного интерфейса IServiceProvider, который определяет механизм для извлечения объекта службы, т. е. объекта, обеспечивающего настраиваемую поддержку для других объектов. [5]

```
Ссылка: 0
public MainViewModel(PageService pageService, MessageBus messageBus)
{
    _pageService = pageService;
    _messageBus = messageBus;

    _pageService.OnPageChanged += PageSourceSetter;
}
```

Рисунок 29 – Конструктор класса MainViewModel с внедрением зависимостей

На рисунке 29 следует обратить внимание на представленный конструктор. Текст «Ссылка: 0» описывает количество активных ссылок на данный конструктор, параметр равен «0», потому что данный конструктор будет вызван неявно вышеописанным сервисом ServiceProvider во время выполнения. В данном случае параметрами конструктора являются два сервиса:

- PageService – сервис, инкапсулирующий в себе событие переключения страниц на объекте Frame;

```
public class PageService
{
    public event Action<Page, IFramedPageViewModel> OnPageChanged;

    Ссылка: 8
    public void ChangePage(Page page, IFramedPageViewModel vm) => OnPageChanged?.Invoke(page, vm);
}
```

Рисунок 30 – Определение класса PageService

- MessageBus – сервис, предоставляющий возможность объектам подписчикам обмениваться указанными объектами, которые наследуют интерфейс IMessage; подписчики определяются классом MessageSubscriber, свойства которого определяют их тип и тип получаемого сообщения; подписчики и ссылки на методы хранятся в потокобезопасной коллекции ConcurrentDictionary _consumers класса MessageBus. Любой другой класс, имеющий ссылку на единственный экземпляр MessageBus может отправлять подписчикам объекты IMessage, если таковые подписаны.

```

public class MessageBus
{
    private readonly ConcurrentDictionary<MessageSubscriber, Func<IMessage, Task>> _consumers;
    Ссылка: 0
    public MessageBus() => _consumers = new();
    Ссылка: 0
    public async Task SendTo<TReceiver>(IMessage message)
    {
        var messageType = message.GetType();
        var receiverType = typeof(TReceiver);
        var tasks = _consumers
            .Where(s => s.Key.MessageType == messageType && s.Key.ReceiverType == receiverType)
            .Select(s => s.Value.Invoke(message));
        await Task.WhenAll(tasks);
    }
    Ссылка: 1
    public void SendTo(IMessage message)
    {
        var messageType = message.GetType();
        _consumers.Where(s => s.Key.MessageType == messageType).ToList().ForEach((x) => x.Value.Invoke(message));
    }
    Ссылка: 1
    public IDisposable Receive<TMessage>(object receiver, Func<IMessage, Task> handler) where TMessage : IMessage
    {
        var subscriber = new MessageSubscriber(receiver.GetType(), typeof(TMessage), s => _consumers.TryRemove(s, out _));
        _consumers.TryAdd(subscriber, (@event) => handler((TMessage)@event));
        return subscriber;
    }
}
Ссылка: 9
public interface IMessage
{
    Ссылка: 5
    public object Object { get; set; }
}
Ссылка: 5
public class MessageSubscriber : IDisposable
{
    private readonly Action<MessageSubscriber> _action;
    Ссылка: 2
    public Type ReceiverType { get; }
    Ссылка: 3
    public Type MessageType { get; }
    Ссылка: 1
    public MessageSubscriber(Type receiverType, Type messageType, Action<MessageSubscriber> action)
    {
        ReceiverType = receiverType; MessageType = messageType; _action = action;
    }
    Ссылка: 0
    public void Dispose() => _action?.Invoke(this);
}

```

Рисунок 30 – Класс MessageBus, MessageSubscriber и интерфейс IMessage

3.6.2 COM, как стандарт для работы с API САПР E3.Series

COM – объектная модель компонентов; – это технологический стандарт от компании Microsoft, предназначенный для создания программного обеспечения на основе взаимодействующих компонентов, каждый из которых может использоваться во многих программах одновременно. Стандарт воплощает в себе идеи полиморфизма и инкапсуляции объектно-ориентированного программирования. Стандарт COM мог бы быть универсальным и платформо-независимым, но закрепился в основном на операционных системах семейства Microsoft Windows. В современных версиях Windows COM используется очень широко. На основе COM были реализованы технологии: Microsoft OLE Automation, ActiveX, DCOM, COM+, DirectX, а также XPCOM [9].

Основным понятием, которым оперирует стандарт COM, является COM-



3.6.3 Взаимодействие разрабатываемого приложения с базой данных

Существуют несколько способов работы с базой данных с использованием языка программирования C#:

- Entity Framework;
- ADO.NET;
- Dapper.

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, где идет работа с объектами [10].

Центральной концепцией Entity Framework является понятие сущности или entity. Сущность представляет набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

Любая сущность, как и любой объект из реального мира, обладает рядом свойств. Например, если сущность описывает человека, то мы можем выделить такие свойства, как имя, фамилия, рост, возраст, вес. Свойства необязательно представляют простые данные типа int, но и могут представлять более комплексные структуры данных. И у каждой сущности может быть одно или несколько свойств, которые будут отличать эту сущность от других и будут уникально определять эту сущность. Подобные свойства называют ключами.

При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

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 [10].

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 [10].

Способы взаимодействия с БД:

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 [10]

• • •

3.6.4 Получение PDF-файлов листов изменений заводской технической документации

Последним этапом в соответствии с моделью логики работы приложения, представленной в главе 2 настоящей ВКР, является «Подготовка PDF-файлов листов изменений ЗТД». В соответствии с техническим заданием (см. Приложение А) программа для получения PDF должна использовать шаблон, представленный в программе MS Office Word (см. Приложение А (к техническому заданию)); работа с данным приложением осуществляется посредством COM.

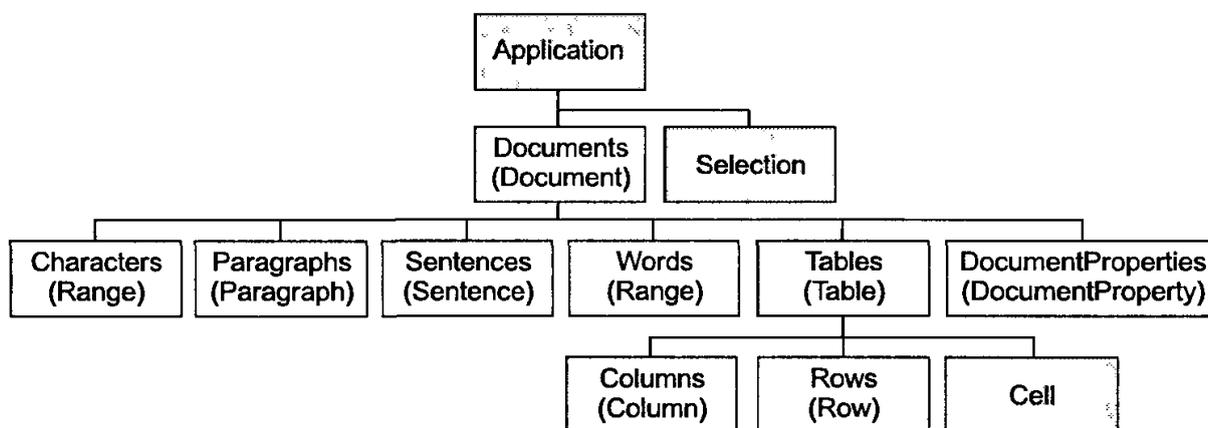


Рисунок 43 – Объектная модель COM Microsoft Office Word [16]

Данная модель указывает, что каждый объект в приложении имеет свое представление в модели COM. На рисунке 44 представлен пример, указывающий на способ получения таблиц и работы с ними.

```
Ссылка: 0
public static void Method()
{
    Word.Application app = new();
    Word.Document doc = app.Documents.Open(@"C:\Users\mds\Desktop\Шаблон ЛИ.docx",
    ReadOnly: false, Visible: false);
    Word.Tables tables = doc.Tables;

    foreach (var tableObject in tables)
    {
        var table = (Word.Table)tableObject;
        foreach (var rowObject in table.Rows)
        {
            var row = (Word.Row)rowObject;
            foreach (var cellObject in row.Cells)
            {
                var cell = (Word.Cell)cellObject;
                if (cell.Range.Text.Trim('\r', '\a')?.Length == 0)
                    cell.Range.Text = "test";
            }
        }
    }

    doc.Save();
    doc.Close();
    app.Quit();
}
```

Рисунок 44 – Получение таблиц в указанном файле и заполнение значениями

В соответствии с этим, полученные таблицы изменений в виде коллекций объектов передаются в работу этому модулю.

Выводы

В результате анализа информации были проведены работы по разработке логической модели работы разрабатываемого продукта, его базы данных на основании данных технического задания: отслеживаемых объектов заводской технической документации. Разработан прототип интерфейса средствами языка разметки XAML, который был согласован с проектным отделом организации. Интерфейс соответствует требованиям заказчика, его техническому заданию и логики работы приложения: 1 этап – Выбор потребителя, 2 этап – Анализ изменений, 3 этап – Подготовка PDF-файлов листов изменений.

Это позволило перейти к этапу разработки системы: база данных, программный интерфейс и внутренняя его логика.

Также было разработано программное обеспечение «Подготовка листов изменений ЗТД»: создание базы данных средствами Microsoft SQL Server Management Studio, написание SQL запросов для получения нужных данных; разработка интерфейса и программной части с описанием используемых сервисов и принципов их разработки с использованием Dependency Injection, описанием работы COM по отношению к САПР E3.Series; описание классов и общей логики приложения.

Приложение прошло опытное тестирование и тестирование членами комиссии по вводу в эксплуатацию, составлен акт ввода в эксплуатацию, представленный в приложении Ж к настоящей ВКР.

ЗАКЛЮЧЕНИЕ

В результате работы была выполнена следующая цель: повышение эффективности процесса подготовки листов изменений заводской технической документации ООО «СЧЕТМАШ» за счет разработки программы, автоматизирующей этот процесс. А также были выполнены следующие задачи:

1. Исследование бизнес-процессов создания сотрудником проектного отдела организации заводской технической документации;
2. Исследование предметной области;
3. Исследование инструментов разработки приложений для персонального компьютера;
4. Написание технического задания на разработку приложения;
5. Прототипирование интерфейса;
6. Проектирование базы данных;
7. Разработка автоматизированной системы;
8. Описание созданной автоматизированной системы.

В результате было разработано программное обеспечение «Подготовка листов изменений ЗТД»: создание базы данных средствами Microsoft SQL Server Management Studio, написание SQL запросов для получения нужных данных; разработка интерфейса и программной части с описанием используемых сервисов и принципов их разработки с использованием Dependency Injection, описанием работы COM по отношению к САПР E3.Series; описание классов и общей логики приложения.

Приложение прошло опытное тестирование и тестирование членами комиссии по вводу в эксплуатацию, составлен акт ввода в эксплуатацию, представленный в приложении Ж к настоящей ВКР.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Официальный сайт ООО «СЧЕТМАШ». О Компании. [Сайт]. – URL: <https://chm.ru/about-the-plant/about/> (дата обращения: 11.05.2022).
2. Махов О.Н. Общее описание САПР E3.Series / О.Н. Махов – Текст : электронный // Современные материалы, техника и технологии. – 2015. №2 (2). – С. 230–233. – URL: <https://cyberleninka.ru/article/n/obschee-opisanie-sapr-e3-series/viewer> (дата обращения: 28.04.2022). – Режим доступа: Научная электронная библиотека Киберленинка.
3. Чухран Д. Иерархическое проектирование в E3.Series / Д. Чухран – Текст: электронный // САПР и графика. – 2009. – С. 61–65. – URL: <https://sapr.ru./article/20020?> (дата обращения: 29.04.2022). – Режим доступа: Научная электронная библиотека eLIBRARY.RU.
4. Dependency Injection. [Сайт]. – URL: METANIT.COM – Сайт о программировании (дата обращения: 27.04.2022).
5. IServiceProvider Интерфейс. [Сайт]. – URL: <https://goo-gl.me/tnYdC> (дата обращения: 28.04.2022).
6. Component Object Model. [Сайт]. – URL: https://microsoft.fandom.com/ru/wiki/Component_Object_Model (дата обращения: 28.04.2022).
7. K. Brockschmidt, Inside OLE 2: the fast track to building powerful object-oriented applications, [Текст] / Brockschmidt К. // Microsoft Press – Redmond, 1994. – P. 1012.
8. K. Brockschmidt, OLE 2.0 Part I: Windows Objects and the Component Object Model, [Текст] / Brockschmidt К. // Microsoft Systems Journal – Redmond, 1993. – P. 23.
9. The Component Object Model: Technical Overview. [Сайт]. – URL: <https://www.cs.umd.edu/~pugh/com/> (дата обращения: 30.04.2022).
10. Введение в Entity Framework. [Сайт]. – URL: <https://metanit.com/sharp/entityframework/1.1.php> (дата обращения: 30.04.2022).
11. Работа с Dapper в ASP.NET Core. [Сайт]. – URL: <https://metanit.com/sharp/aspnet5/26.1.php> (дата обращения: 30.04.2022).

12. Кинзябулатов Р. Моделирование бизнеса. Основные подходы / Кинзябулатов Р. – Текст: электронный // Бизнес модели. – 2017. – URL: <https://habr.com/ru/company/trinion/blog/332772/> (дата обращения: 28.04.2022).
13. NET Framework, Библиотеки виджетов. Windows Forms. [Сайт]. – URL: https://microsoft.fandom.com/ru/wiki/Windows_Forms (дата обращения: 28.04.2022).
14. Разработка информационно-аналитической системы. [Сайт]. – URL: https://studbooks.net/2156571/informatika/tehnologii_entity_framework_adonet? (дата обращения: 28.04.2022).
15. Петров С. Паттерны проектирования для новичков: MVC vs MVP vs MVVM / С. Петров – Текст: электронный // Паттерны проектирования. – 2014. – URL: <https://habr.com/ru/post/215605/?> (дата обращения: 30.04.2022).
16. Общие сведения об объектной модели Word. [Сайт]. – URL: <https://docs.microsoft.com/ru-ru/visualstudio/vsto/word-object-model-overview?view=vs-2019> (дата обращения: 28.04.2022).
17. Официальный сайт Visual Studio. [Сайт]. – URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 28.04.2022).
18. SQL Server Management Studio. [Сайт]. – URL: <https://docs.microsoft.com/ru-ru/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15> (дата обращения: 28.04.2022).
19. Официальный сайт JetBrains. [Сайт]. – URL: <https://www.jetbrains.com/rider/> (дата обращения: 28.04.2022).

ПРИЛОЖЕНИЕ А

Техническое задание на разработку автоматизированной системы
МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Удмуртский государственный университет»

Филиал федерального государственного бюджетного образовательного учреждения
высшего образования
«Удмуртский государственный университет» в городе Воткинске

Кафедра информационных и инженерных технологий

09.03.03 «Прикладная информатика»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Тема: «Разработка системы для автоматизации процесса подготовки листов
изменений заводской технической документации ООО «СЧЕТМАШ»

Согласовано:

Исполнитель
студент группы О ОБ-Вт-090303-41

И.И. Иванов

Руководитель
преп., канд. тех. наук

П.П. Петров

Воткинск 2022

СОДЕРЖАНИЕ

Введение	3
1 Основание для разработки	3
2 Назначение разработки	3
3 Требования к проекту	3
3.1.Требования к функциональным характеристикам	3
3.2.Требования к надежности и безопасности	4
3.3.Условия эксплуатации.....	4
3.4.Требования составу и параметрам технических средств.....	4
3.5.Требования к информационной и программной совместимости.....	5
3.6.Требования к интерфейсу системы	5
3.7.Требования к группам пользователей.....	7
4 Требования к программной документации	8
5 Стадии и этапы разработки	8
6 Порядок контроля и приемки.....	8

ВВЕДЕНИЕ

Настоящее Техническое задание определяет требования и порядок разработки автоматизированной системы заказчика Проектный отдел ООО «СЧЕТ-МАШ» далее Заказчик.

Наименование программы – «Подготовка листов изменений заводской технической документации» далее «Подготовка листов изменений ЗТД».

Краткая характеристика области применения программы – область применения автоматизированной системы заказчика достаточно узкая. Данная система рассчитана для внедрения в бизнес-процесс подготовки заводской технической документации (далее ЗТД) к выдаче заказчикам и производству завода.

1 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основание для разработки - разработка ведется на основании приказа №17/02-08 от 29.03.2022 о закреплении тем выпускных квалификационных работ за студентами, а также на основании заявки заказчика №000000098 от 15.06.2021.

2 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Автоматизированная система заказчика «Подготовка листов изменений ЗТД» предназначена предоставлять возможность сотрудникам заказчика подготавливать PDF-файлы листов изменений ЗТД, а также предпросмотр изменений, внесенных в текущую версию проекта относительно предыдущей.

3 ТРЕБОВАНИЯ К ПРОЕКТУ

3.1. Требования к функциональным характеристикам

Система должна обеспечивать возможность выполнения перечисленных ниже функций:

- Сравнение предыдущей и текущей версии ЗТД;
- Предпросмотр изменений;
- Подготовка листов изменений ЗТД;
- Просмотр информации о фактах выдачи ЗТД потребителям.

3.2. Требования к надежности и безопасности

Надежное (устойчивое) функционирование системы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- Бесперебойный доступ к СУБД;
- Система, имеющая характеристики, описанные в пункте 3.4.;
- Регулярное создание резервных копий баз данных.

3.3. Условия эксплуатации

Минимальное количество людей, требуемых для работы системы, должно составлять не менее 2 единиц – системный программист и конечный пользователь программы – сотрудник проектного отдела.

Системный программист должен иметь минимум среднее техническое образование в области управления СУБД и программировании.

В перечень задач, выполняемых системным программистом, должны входить:

- поддержание работоспособности системы;
- устранение возникших проблем на сервере и в текущей системе;

Пользователи должны уверенно понимать предметную область.

3.4. Требования к составу и параметрам технических средств

Программа должна корректно отображаться и работать на устройствах с горизонтальным разрешением экрана не менее 720 пикселей. Работа на устройствах, не соответствующих заявленным требованиям, возможна, но не гарантирована.

В состав технических средств должен входить компьютер или ноутбук.

Компьютер/ноутбук должен включать в себя следующие характеристики:

- Разрешение экрана не менее 1024x768;
- Наличие клавиатуры или иного средства ввода данных;
- Оптический манипулятор типа «мышь» или тачпад;
- Не менее 1 Гб оперативной памяти.

3.5. Требования к информационной и программной совместимости

Работа обеспечивается серверными и клиентскими компонентами.

Требования к серверным компонентам:

- MS SQL Server 2012 и выше;
- Сервер лицензий САПР E3.Series;
- Операционная система Windows Server 2012 и выше.

Требования к клиентским компонентам:

- Операционная система Windows 7 и выше;
- САПР E3.Series и её компоненты для проектирования;
- Соответствующие среды выполнения программы.

3.6. Требования к интерфейсу системы

Внешний интерфейс программы должен быть масштабируемым и выглядеть одинаково на разных разрешениях экрана

Требуемые поддерживаемые разрешения экрана:

- Full-HD 1920 x 1080 пикселей

Информация, отображаемая в программе, должна помещаться в область интерфейса программы. Элементы интерфейса и сам интерфейс должны быть выполнены в едином стиле без использования полной насыщенности цветов и оттенков.

Требования к цветам, используемым в программе (далее требования к оформлению):

- Основной цвет текста #000000 (Черный);
- Дополнительный цвет текста #363636 (Не насыщенный черный, темно-серый);
- Цвет фона элементов (кроме кнопок) #F5F5F5 (Теплый белый с оттенками серого);
- Цвет границ элементов (кроме кнопок) #CBCBCB (Теплый серый);
- Цвет фона выделенных элементов #DEDEDE (Темно-серый);
- Цвет текста кнопок, не расположенных на панели инструментов #607D8B (Холодный серо-синий);
- Цвет границ кнопок, не расположенных на панели инструментов #607D8B (Холодный серо-синий);

- Цвет текста кнопок, расположенных на панели инструментов #FFFFFF (Белый);
- Цвет границ кнопок, расположенных на панели инструментов #8197A3 (Светлый холодный серо-синий);
- Цвет панели инструментов #607D8B (Холодный серо-синий);
- Цвет иконки «Предупреждение» диалогового окна #EA9801 (Не насыщенный оранжевый);
- Цвет иконки «Информация» диалогового окна #5790CB (Не насыщенный темно-синий);
- Цвет иконка «Ошибка» диалогового окна #AA0000 (Не насыщенный красный).

Требования к оформлению таблиц:

- Использование цветов в соответствии требованиями оформления;
- Информация в ячейках таблиц должна быть читаемой.

Требования к оформлению диалоговых окон:

- Использование цветов в соответствии с требованиями оформления;
- Диалоговые окна должны иметь значимость в используемом контексте;
- Диалоговые окна должны подразделяться на виды (ошибка, предупреждение, информация);
- Пользователь должен иметь возможность отменить действие, которое породило диалоговое окно.

Система должна иметь 2 вкладки:

- Сравнение изменений;
- Подготовка листов изменений.

Вкладка сравнения изменений должна содержать следующее:

- Таблицы сравнения отслеживаемых объектов ЗТД;
 - а. соединениях и их адресации.

Данные объекты должны быть выведены в таблицы-сравнения предыдущей и текущей версии проекта для каждого объекта.

- Кнопка «Сравнить».

Вкладка подготовка листов изменений:

— Таблица с информацией обо всех версиях ЗТД, для которых готовились листы изменений;

— Таблицы (для каждого потребителя) с информацией обо всех фактах передачи ЗТД потребителю.

Таблицы должны иметь поля: Дата формирования ЗТД, пользователь, который подготовил ЗТД и описание изменений.

3.7. Требования к группам пользователей

Пользователи продукта – сотрудники проектного отдела, которые не имеют учетных записей в составе данной разрабатываемой программы, доступ к базе данных, используемой программой имеют сотрудники отдела информационных технологий.

4 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации должен включать в себя:

— Техническое задание;

— Описание системы;

— Текст программы;

— Пояснительная записка.

5 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Работы, предусмотренные настоящим техническим заданием, должны быть выполнены в 8 этапов:

1. Анализ и описание предметной области;
2. Определение основных требований для системы;
3. Прототипирование интерфейса пользователя;
4. Разработка интерфейса пользователя;
5. Разработка внутренней составляющей системы;
6. Тестирование системы;
7. Подготовка программной документации;
8. Ввод в эксплуатацию.

6 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком, в лице начальника Проектного отдела, программы и методик испытаний, проверяется выполнение всех функций программы. Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний.

ПРИЛОЖЕНИЕ Б

Форма таблицы изменений соединений

Таблица А.1 – Форма таблицы изменений соединений

<i>Изменение от</i>				<i>Дата-время</i>															
<i>Соединение удалено</i>																			
<i>было</i>										<i>стало</i>									
<i>откуда</i>				<i>куда</i>				цепь	провод/ кабель	<i>откуда</i>				<i>куда</i>				цепь	провод/ кабель
устройство	место	изделие	клемма	устройство	место	изделие	клемма			устройство	место	изделие	клемма	устройство	место	изделие	клемма		
<i>Соединение добавлено</i>																			
<i>было</i>										<i>стало</i>									
<i>откуда</i>				<i>куда</i>				цепь	провод/ кабель	<i>откуда</i>				<i>куда</i>				цепь	провод/ кабель
устройство	место	изделие	клемма	устройство	место	изделие	клемма			устройство	место	изделие	клемма	устройство	место	изделие	клемма		
<i>Соединение изменено</i>																			
<i>было</i>										<i>стало</i>									
<i>откуда</i>				<i>куда</i>				цепь	провод/ кабель	<i>откуда</i>				<i>куда</i>				цепь	провод
устройство	место	ПО	клемма	устройство	место	ПО	клемма			устройство	место	ПО	клемма	устройство	место	ПО	клемма		

ПРИЛОЖЕНИЕ Г

Классы-репозитории, содержащие описание конкретных экземпляров классов
с отдельным жизненным циклом.

Класс-репозиторий, содержащий описание экземпляров интерфейсов
приложения:

```
public class ViewRepository
{
    private static ServiceProvider _provider;

    public static SelectConsumerPageView SelectConsumerPage =>
_provider.GetRequiredService<SelectConsumerPageView>();

    public static AnalysisChangesPageView AnalysisChangesPage =>
_provider.GetRequiredService<AnalysisChangesPageView>();

    public static ConnectionsPageView ConnectionsPage => _provider.GetRequiredService<ConnectionsPageView>();

    public static PreparingPdfPageView PreparingPdfPage => _provider.GetRequiredService<PreparingPdfPageView>();

    public static ProjectTransferStoryPageView ProjectTransferStoryPage =>
_provider.GetRequiredService<ProjectTransferStoryPageView>();

    public static DialogWindow DialogWindow => _provider.GetRequiredService<DialogWindow>();
```

• • •

Класс-репозиторий, содержащий описание экземпляров *ViewModel*

приложения:

```
public class ViewModelRepository
{
    private static ServiceProvider _provider;
```

• • •

ПРИЛОЖЕНИЕ Д

Объектная модель SOM для API САПР E3.Series

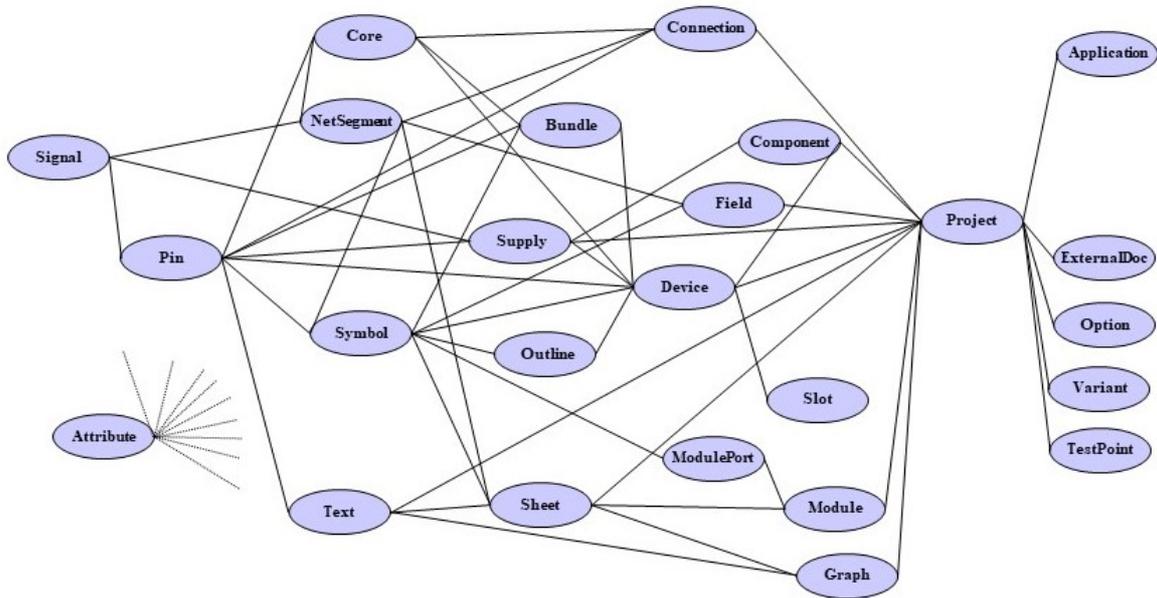


Рисунок Г.1 – Объектная модель SOM

ПРИЛОЖЕНИЕ Е

Запросы SQL для работы с фреймворком Dapper

Запросы представлены в виде статических строковых полей класса DBQuery. Описание данных запросов представлено в комментариях кода.

```
internal static class DBQueries
{
    //Запрос на получение последней версии проекта в таблице Versions
    internal static string _getLatestVersionQuerySql = @"select max(id) from Versions";

    //Запрос на получение всех потребителей из таблицы Consumers
    //Используется при запуске приложения и выводится на страницу SelectConsumerPageView
    //в соответствующий ComboBox
    internal static string _getConsumersQuerySql = @"select Id, Name from Consumers";
```

• • •

Учебное издание

Составители:

Мамрыкин Олег Владимирович, Кузнецов Андрей Павлович,
Раскин Павел Николаевич

**Методические рекомендации по подготовке и защите выпускных
квалификационных работ по направлению подготовки 09.03.03 «Приклад-
ная информатика», профиля подготовки «Прикладная информатика в
экономике»**

Авторская редакция

Компьютерная верстка: Ю. Н. Небрачных

Издательский центр «Удмуртский университет»
426034, г. Ижевск, ул. Ломоносова, 4Б, каб. 021
Тел. : + 7 (3412) 916-364, E-mail: editorial@udsu.ru