

Оглавление

Введение	2
Глава 1. Теоретические основы разработки требований к программной системе	5
1.1. Требования к программной системе и обоснование их разработки	5
1.2. Пользовательские требования	7
1.3. Функциональные и нефункциональные требования	8
Глава 2. Предпроектное исследование и обоснование необходимости разработки ПС	10
2.1. Анализ предметной области	10
2.2. Сравнительный анализ существующих решений	13
2.3. Моделирование целевых бизнес-процессов и концепция продукта	16
2.4 Концепции продукта и границы проекта	23
Глава 3. Разработка требований к программной системе	25
3.1. Определение и документирование пользовательских требований	25
3.2. Определение и документирование функциональных требований	28
3.3. Определение и документирование нефункциональных требований	37
Заключение	41
Список используемых источников	43
Приложения	44
Приложение 1. Документ о концепции и границах	44
Приложение 2. Атрибуты качества	50

Введение

В современных условиях высокой конкуренции эффективность управления сбытом готовой продукции является критическим фактором успеха для производственных предприятий. Скорость обработки заказов, точность контроля складских остатков, оперативность логистики и качество взаимодействия с клиентами напрямую определяют финансовые результаты и рыночную репутацию компании. Тем не менее, на многих предприятиях ключевые процессы сбыта до сих пор строятся на рутинных, слабо автоматизированных операциях: учет в электронных таблицах (Excel), ручное формирование документов, разрозненное хранение информации о клиентах и заказах, обмен данными между отделами посредством электронной почты и телефонных звонков.

Подобный подход порождает ряд системных проблем: дублирование и порча данных, высокие временные затраты на оформление одной сделки, ошибки при резервировании товара, задержки в отгрузках, сложности с формированием оперативной аналитики для принятия управленческих решений. Всё это приводит к потерям клиентов, неэффективному использованию складских площадей и, как следствие, к снижению общей прибыльности бизнеса.

В связи с этим возникает острая потребность в разработке и внедрении специализированной программной системы, которая позволила бы автоматизировать полный цикл управления сбытом – от приема заказа до отгрузки и постпродажного анализа. Грамотно разработанные требования к такой системе служат фундаментом успешного проекта, минимизируя риски несоответствия конечного продукта ожиданиям бизнеса и обеспечивая его интеграцию в существующую ИТ-инфраструктуру предприятия.

Объект исследования – ООО «Дистрибьютер экспресс»

Предмет исследования – управление сбытом готовой продукции

Цель работы – обоснование необходимости создания и разработка полного комплекса требований (бизнес, пользовательских, функциональных и нефункциональных) к программной системе «SalesControl».

Задачи:

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить теоретические основы разработки и управления требованиями к программному обеспечению.
2. Провести предпроектное исследование предметной области: проанализировать текущие бизнес-процессы сбыта («как есть») и выявить их ключевые недостатки.
3. Обосновать необходимость разработки новой системы и смоделировать целевые бизнес-процессы («как должно быть»), определив границы проекта.
4. Определить и задокументировать пользовательские требования, разработав диаграмму вариантов использования и набор пользовательских историй.
5. Разработать функциональные требования, спроектировав логику и структуру системы с помощью диаграмм UML (классов, последовательности, деятельности) и модели «сущность-связь».
6. Сформулировать нефункциональные требования к системе, включая атрибуты качества, ограничения, описание интерфейсов и архитектуры (диаграммы компонентов и развертывания).
7. Разработать техническое задание (ТЗ) на создание системы.

В соответствии с поставленной целью и задачами структура работы состоит из введения, трёх глав, заключения, списка использованной литературы и двух приложений.

Введение обосновывает актуальность темы, определяет объект, предмет, цель и задачи исследования.

Первая глава посвящена изучению теоретических основ разработки требований к программному обеспечению, включая классификацию требований (бизнес-, пользовательские, функциональные и нефункциональные) и методы их документирования.

Во второй главе проводится предпроектное исследование предметной области: анализируется текущее состояние бизнес-процессов сбыта, выявляются их недостатки, обосновывается необходимость разработки новой системы, моделируются целевые процессы и формулируется концепция продукта.

Третья глава содержит непосредственную разработку требований к системе «SalesControl»: пользовательских (диаграмма вариантов использования, истории), функциональных (UML-диаграммы классов, последовательности, деятельности, ER-модель) и нефункциональных (атрибуты качества, описание архитектуры и интерфейсов).

Заключение подводит итоги работы, формулирует основные выводы и намечает перспективы дальнейших исследований.

Список использованной литературы включает источники, на которые опирался автор в процессе исследования.

В Приложении 1 представлен Документ о концепции и границах проекта (Scope & Vision Document), а в Приложении 2 — детализированный перечень атрибутов качества системы.

Глава 1. Теоретические основы разработки требований к программной системе

1.1. Требования к программной системе и обоснование их разработки

Разработка программного обеспечения (ПО) – это сложный инженерный процесс, успех которого в решающей степени определяется качеством работы на начальных, аналитических этапах. Согласно исследованиям и статистике IT-индустрии, одной из главных причин провала проектов, превышения бюджета или сроков является неполное, противоречивое или постоянно меняющееся множество требований.

Требование – это условие или возможность, которой должна обладать система, чтобы удовлетворять контракту, стандарту, спецификации или потребностям пользователя и других заинтересованных сторон (стейкхолдеров).

Процесс инженерии требований (Requirements Engineering) представляет собой систематический и поддающийся анализу подход к выявлению, документированию, анализу и проверке требований. Он включает в себя несколько ключевых итеративных этапов:

1. **Выявление (Elicitation):** Сбор информации от всех заинтересованных лиц (менеджеров, конечных пользователей, клиентов, технических специалистов) с помощью интервью, анкетирования, мозговых штурмов, анализа документации и наблюдения за рабочими процессами.
2. **Анализ (Analysis):** Структурирование и осмысление собранной информации, выявление противоречий, расстановка приоритетов, уточнение деталей и построение моделей для лучшего понимания предметной области.
3. **Спецификация (Specification):** Формальная или неформальная фиксация требований в виде документов, которые будут использоваться в дальнейшей разработке. Основным документом является Спецификация

требований к программному обеспечению (Software Requirements Specification, SRS).

4. Верификация и валидация (Verification & Validation): Проверка требований на корректность, полноту, непротиворечивость, реализуемость и тестируемость, а также подтверждение того, что разработанный набор требований действительно решает проблемы пользователей и соответствует бизнес-целям.

Классификация требований по уровням детализации и целевой аудитории является общепринятой практикой:

1. Бизнес-требования (Business Requirements): описывают высокоуровневые цели организации, которую система должна поддерживать. Отвечают на вопрос «*Зачем нужна система? Какую выгоду она принесет?*» (например, «сократить время обработки заказа на 40%»).
2. Пользовательские требования (User Requirements): описывают задачи, которые пользователи смогут выполнять с помощью системы. Отвечают на вопрос «*Что пользователь хочет делать?*» (например, «менеджер хочет оформить заказ»). Часто документируются в виде пользовательских историй и диаграмм вариантов использования.
3. Функциональные требования (Functional Requirements): описывают конкретное поведение системы, ее функции и реакции на входные данные. Отвечают на вопрос «*Что система должна делать?*» (например, «система должна резервировать товар на складе при создании заказа»).

Обоснование тщательной разработки требований заключается в необходимости создания единого, разделяемого всеми участниками проекта понимания будущего продукта. Этот процесс минимизирует риск дорогостоящих переделок на поздних этапах разработки и тестирования, обеспечивает основу для планирования, служит критерием для приемки

системы и, в конечном счете, является ключевым фактором успешной реализации ИТ-проекта.

1.2. Пользовательские требования

Пользовательские требования (User Requirements) служат мостом между высокоуровневыми бизнес-целями и детальными техническими спецификациями. Они фокусируются на целях пользователя, описывая, какие задачи он должен иметь возможность решать с помощью системы, не углубляясь в механизмы реализации.

Для эффективного описания и анализа пользовательских требований используются следующие инструменты:

1. Диаграммы вариантов использования (Use Case Diagrams) в нотации UML. Это графическое представление функциональности системы с точки зрения внешних взаимодействий.
 - Актеры (Actors): Роли, которые пользователи или внешние системы играют по отношению к системе (например, «Менеджер», «Кладовщик», «Система 1С»).
 - Варианты использования (Use Cases): отдельные целостные функции, выполняемые системой для достижения цели актера (например, «Оформить заказ», «Зарезервировать товар»).
 - Связи (Relationships): определяют отношения между актерами и вариантами использования (ассоциация), а также между самими вариантами использования (включение, расширение, обобщение).
2. Пользовательские истории (User Stories). Это краткие, сфокусированные на ценности формулировки требований на естественном языке. Классический шаблон пользовательской истории: «Как [роль], я хочу [действие/цель], чтобы [ценность/результат]».
 - Пример: «Как менеджер, я хочу видеть актуальные остатки товаров на складе в режиме реального времени, чтобы сразу подтверждать клиенту возможность заказа и избегать ситуации с отсутствующим товаром».

- Пользовательские истории удобны для приоритизации, обсуждения с заказчиком и разбивки на более мелкие технические задачи.

Документирование пользовательских требований в таком виде позволяет четко определить границы системы, убедиться, что она будет полезна и удобна для конечных пользователей, и создать основу для детального проектирования.

1.3. Функциональные и нефункциональные требования

После определения того, *что* нужно пользователю, необходимо детально описать, *как* система будет это реализовывать. Для этого разрабатываются функциональные и нефункциональные требования.

Функциональные требования (Functional Requirements) определяют конкретные поведения, действия, вычисления и реакции системы. Они отвечают на вопрос «*Что система должна делать?*» и описывают:

- Обработку входных данных (например, валидация данных заказа).
- Выполняемые вычисления и бизнес-логику (например, расчет итоговой суммы с учетом скидок).
- Манипуляции с данными (CRUD-операции: создание, чтение, обновление, удаление).
- Взаимодействие с другими системами (например, отправка данных в бухгалтерскую систему).
- Формирование выходных данных (отчеты, документы).

Для моделирования функциональных требований широко используется язык UML:

- Диаграмма классов (Class Diagram): показывает статическую структуру системы: классы, их атрибуты, методы и взаимосвязи (ассоциации, наследование, агрегация).
- Диаграмма последовательности (Sequence Diagram): демонстрирует взаимодействие объектов во времени, порядок обмена сообщениями для выполнения конкретного сценария.

- Диаграмма деятельности (Activity Diagram): описывает логику процессов и алгоритмов, включая ветвления, параллельные потоки и циклы. Фактически моделирует бизнес-процесс с точки зрения системы.
- Диаграмма «сущность-связь» (ER-diagram): используется для проектирования структуры базы данных, отображает сущности, их атрибуты и связи между ними.

Нефункциональные требования (Non-functional Requirements), или атрибуты качества, описывают свойства системы, не связанные напрямую с её конкретными функциями, но критически важные для её эксплуатации и восприятия. Они отвечают на вопрос *«Как хорошо система должна это делать?»* и включают:

1. Производительность (Performance): Время отклика, пропускная способность, использование ресурсов.
2. Надежность (Reliability): Устойчивость к сбоям, среднее время наработки на отказ (MTBF), точность выполнения операций.
3. Безопасность (Security): Защита от несанкционированного доступа, целостность и конфиденциальность данных, аутентификация и авторизация.
4. Удобство использования (Usability): Интуитивность интерфейса, время обучения, эффективность работы пользователя.
5. Совместимость (Compatibility): Способность взаимодействовать с другим программным обеспечением и аппаратными платформами.
6. Масштабируемость (Scalability): Возможность увеличения производительности при росте нагрузки.
7. Поддерживаемость (Maintainability): Легкость анализа, изменения, тестирования и исправления системы.

Совокупность функциональных и нефункциональных требований формирует техническое задание (ТЗ), которое является основным договорным и техническим документом, регламентирующим процесс создания программной системы.

Глава 2. Предпроектное исследование и обоснование необходимости разработки ПС

2.1. Анализ предметной области

Описание текущего положения

Объектом исследования является служба ООО «Дистрибьютер экспресс» автоматизации выступают процессы отдела продаж и сбыта типичного производственного предприятия малого или среднего масштаба.

В настоящий момент ключевые операции по управлению сбытом готовой продукции выполняются вручную с использованием разрозненных, не интегрированных инструментов. Как показало предпроектное обследование (отчет «As-Is»), основные бизнес-процессы строятся следующим образом:

1. Прием и оформление заказа: менеджер получает заявку от клиента по телефону или электронной почте. Для проверки наличия товара он обращается к кладовщику устно или через мессенджер, либо сверяется с устаревшим Excel-файлом остатков. Коммерческое предложение и счет формируются вручную в MS Word и Excel на основе шаблонов, что занимает значительное время и чревато ошибками в реквизитах и ценах.
2. Учет клиентов и история взаимодействий: информация о клиентах (контакты, история заказов) хранится в персональных Excel-таблицах каждого менеджера, а также в почтовом клиенте и записных книжках. Единой клиентской базы не существует, что приводит к потере информации при уходе сотрудника и невозможности коллегиальной работы с клиентом.
3. Складской учет и резервирование: кладовщик ведет учет приходов и расходов в бумажном журнале или в локальном файле Excel. Резервирование товара под заказ часто происходит «на словах», что приводит к двойным продажам одного и того же остатка («перепродажа»). Актуальный остаток известен только на момент инвентаризации.

4. Организация отгрузки и доставки: после комплектации заказа кладовщик вручную заполняет бумажную товарно-транспортную накладную (ТТН). Менеджер или логист по телефону ищет курьера или транспортную компанию, затем передает им данные о доставке.
5. Формирование отчетности и аналитика: В конце отчетного периода (день, неделя, месяц) руководитель отдела или директор запрашивает у менеджеров данные об их продажах. Менеджеры вручную сводят цифры из своих файлов, что занимает несколько часов. Полученные данные консолидируются в общий отчет, достоверность которого сложно проверить. Анализ эффективности, рентабельности и динамики продаж носит эпизодический и трудоемкий характер.

Основные проблемы текущего состояния:

1. Высокие временные затраты: каждая операция (оформление заказа, согласование, отчет) требует значительных ручных усилий и многочисленных согласований.
2. Критическая зависимость от человеческого фактора: ошибки при вводе данных, забытые резервирования, потеря писем или файлов — обычное явление, напрямую влияющее на финансовые потери и репутацию.
3. Отсутствие единого информационного пространства: данные раздроблены между людьми, файлами и каналами связи. Невозможно оперативно получить целостную картину по заказу, клиенту или складу.
4. Низкая оперативность и прозрачность: статус заказа неизвестен, клиенты вынуждены постоянно звонить с вопросами. Руководство не имеет инструментов для реального контроля и оперативного управления.
5. Сложность масштабирования: с ростом числа заказов и клиентов накладные расходы на ручное управление растут нелинейно, становясь тормозом для развития бизнеса.

Организационная структура и классы пользователей

Для выявления ключевых пользователей будущей системы была проанализирована организационная структура предприятия. Основные роли, задействованные в процессе сбыта, представлены на рисунке 1.

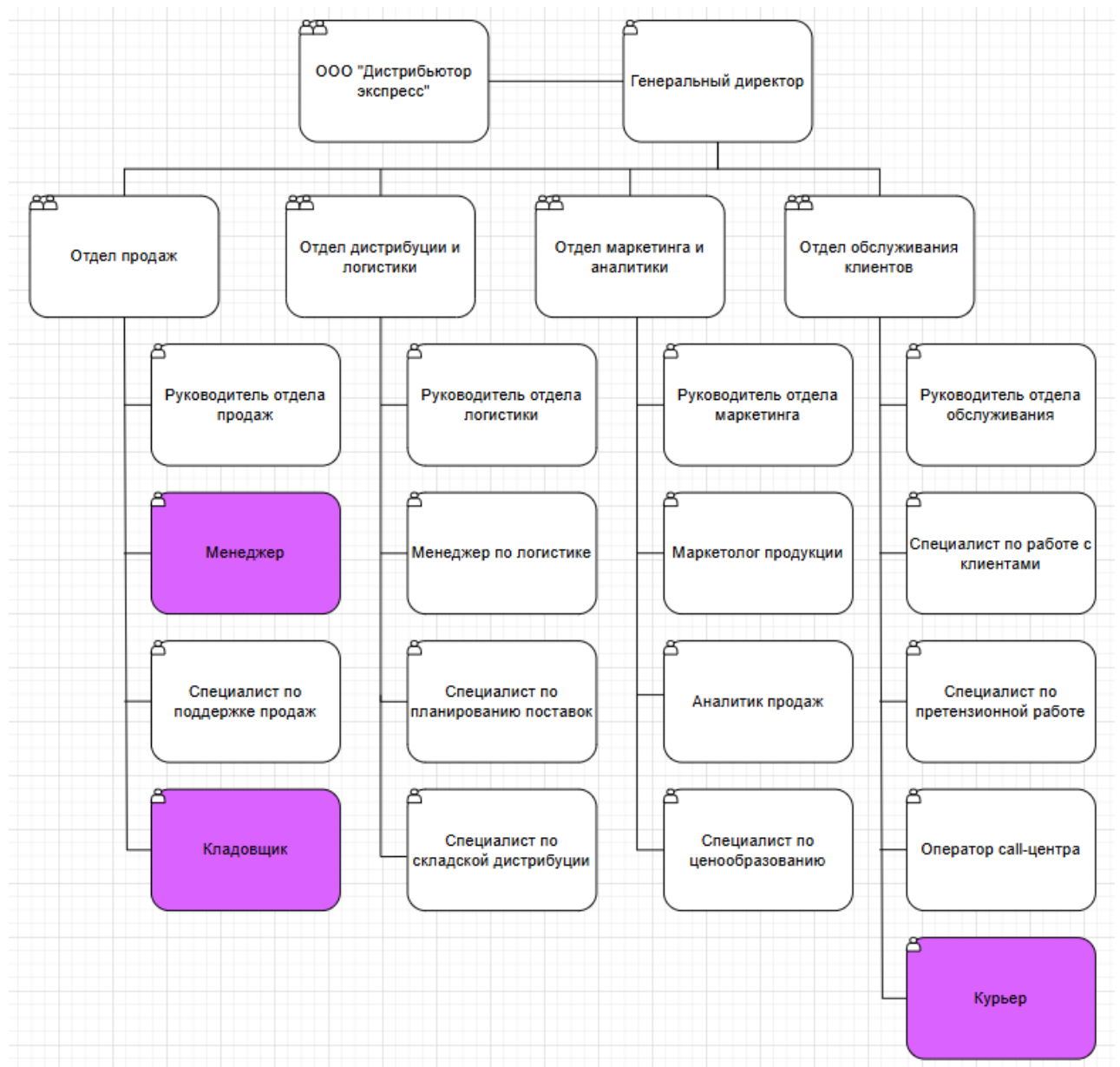


Рисунок 1 – Организационная диаграмма

Рисунок отображает иерархическую организационную структуру ООО «Дистрибьютор экспресс». Анализ структуры позволил выделить ключевых сотрудников (актеров), чья деятельность будет автоматизирована системой «SalesControl»: Менеджера (Отдел продаж), Кладовщика и Курьера (Отдел обслуживания клиентов)

На основе этой структуры выделены следующие классы пользователей (актеры) системы:

Таблица 1 – Классы пользователей системы

№	Класс пользователя	Описание и роль
1	Менеджер	Основной пользователь. Задачи: работа с клиентами, оформление заказов, проверка остатков, контроль статусов заказов, формирование коммерческих предложений.
2	Кладовщик	Ключевой пользователь складского модуля. Задачи: прием и учет товара, комплектация заказов по заданию системы, обновление фактических остатков, формирование отгрузочных документов.
3	Курьер	Пользователь модуля доставки. Задачи: получение заданий на отгрузку, фиксация факта доставки или вывоза товара, ввод данных транспортной компании.

2.2. Сравнительный анализ существующих решений

Для наглядного обоснования необходимости разработки собственного решения проведен сравнительный анализ двух состояний: существующего (ручное управление) и целевого (с внедрением системы «SalesControl»).

Критерии сравнения

Для анализа выбраны критерии, которые являются критически важными для эффективности и конкурентоспособности бизнеса в сфере сбыта:

1. Скорость обработки заказа: время от получения заявки до формирования заданий для склада и логистики.
2. Точность данных: уровень ошибок в документах, при резервировании товара, в финансовых расчетах.

3. Прозрачность процесса: возможность для клиента и сотрудников в любой момент узнать актуальный статус любого заказа.
4. Оперативность отчетности: время, необходимое для формирования сводного отчета о продажах за период.
5. Масштабируемость: способность бизнес-процесса обрабатывать возрастающий поток заказов без пропорционального роста издержек и ошибок.
6. Контроль и управление: наличие у руководства инструментов для мониторинга ситуации и оперативного принятия решений.

Сравнительная таблица

Таблица 2 – Сравнительный анализ систем автоматизации

Критерий	Текущее состояние	Целевое состояние	Эффект от автоматизации
Скорость обработки заказа	20–60 минут (звонки, поиск в Excel, ручной расчет, отправка документов).	3–5 минут (выбор из каталога, автоматический расчет, мгновенное резервирование, печать документов).	Ускорение в 4–10 раз. Сокращение времени реакции на запрос клиента.
Точность данных	Высокий риск ошибок (человеческий фактор при переносе данных, устаревшие прайсы, ошибки в реквизитах).	Минимизация ошибок. Валидация данных системой, актуальные справочники, автоматический расчет.	Снижение ошибок >90%. Сокращение финансовых и репутационных потерь.

Прозрачность процесса	Нулевая. Статус заказа известен только менеджеру, его приходится уточнять по цепочке.	Полная. Каждый этап (создан, оплачен, собран, отгружен, доставлен) фиксируется в системе и доступен уполномоченным лицам.	Повышение клиентской лояльности, снижение нагрузки на поддержку.
Оперативность отчетности	От нескольких часов до дня (ручной сбор и консолидация данных из разных источников).	В реальном времени. Статистика и пред заданные отчеты формируются системой автоматически по клику.	Мгновенная аналитика для принятия управленческих решений.
Масштабируемость	Крайне низкая. Рост заказов ведет к хаосу, потерям и необходимости резкого увеличения штата.	Высокая. Система берет на себя рутину, позволяя существующей команде обрабатывать большой объем без потери качества.	Возможность роста бизнеса без пропорционального роста операционных издержек.
Контроль и управление	Отсутствует. Руководство получает искаженную и запаздывающую	Централизованный дашборд. Наглядное отображение KPI, воронки продаж, загрузки	Управление на основе актуальных данных.

	ю информацию.	склада, эффективности менеджеров.	
--	------------------	---	--

Обоснование актуальности разработки

Разработка собственной системы «SalesControl» экономически обоснована, так как единовременные инвестиции окупятся за счет отказа от дорогих лицензий и сокращения операционных издержек через автоматизацию ручных процессов. Технически она целесообразна, потому что позволит создать гибкий инструмент, идеально адаптированный под уникальные бизнес-регламенты компании, а не подстраиваться под жесткие рамки готовых решений.

Это даст прямое улучшение ключевых показателей: ускорит обработку заказов, повысит точность данных и обеспечит прозрачность процессов для клиентов и руководства.

2.3. Моделирование целевых бизнес-процессов и концепция продукта

Модель «Как есть» (As-Is)

Для детального анализа проблем была построена контекстная диаграмма процессов в текущем состоянии. Диаграмма представлена на рисунках 2–3. Процесс характеризуется ручным управлением и отсутствием автоматизации.

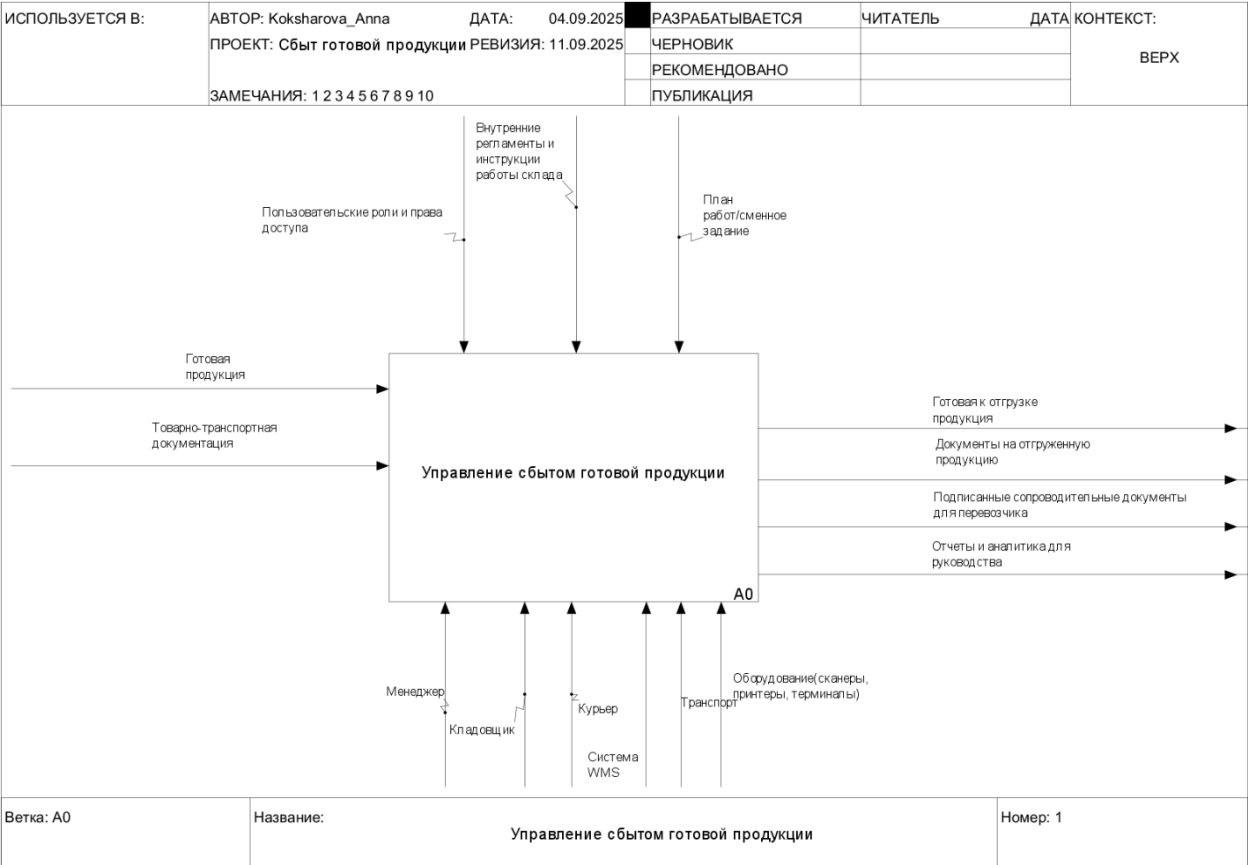


Рисунок 2 – Контекстная диаграмма (A0)

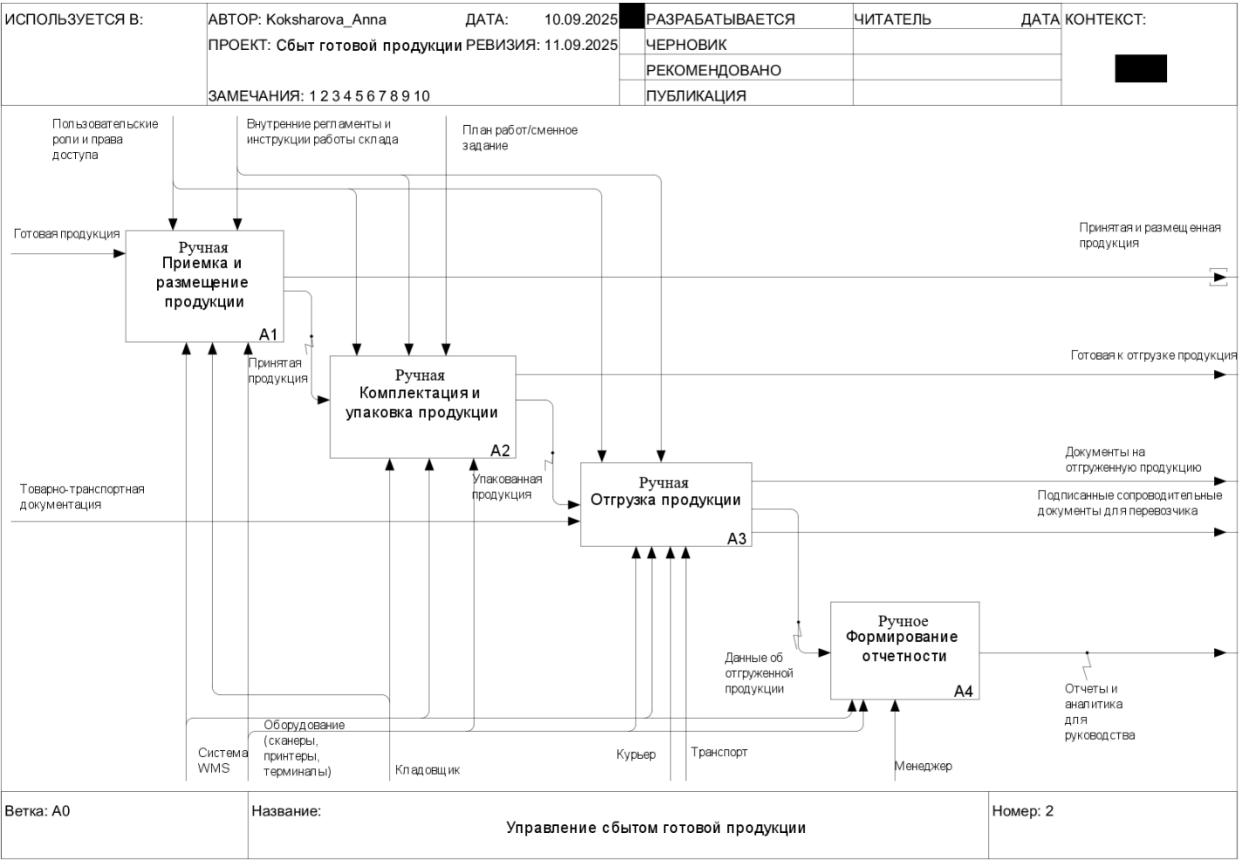


Рисунок 3 – Детализация блока A0

Анализ диаграммы показал наличие «узких мест»: ручной ввод данных в Excel, потеря информации при передаче между сотрудниками, задержки в формировании отчетности.

Определение бизнес-потребностей

Главная цель: повысить эффективность и прибыльность управления сбытом.

- Цель 1: автоматизировать ключевые рутинные операции.
 - Сократить время оформления заказа до 5 мин.
 - Исключить ручной пересчет остатков.
 - Автоматизировать формирование типовых документов.
- Цель 2: обеспечить целостность и актуальность данных.
 - Создать единую базу клиентов и товаров.
 - Внедрить онлайн-резервирование товаров.
 - Вести полную историю по каждому заказу.
- Цель 3: повысить прозрачность и управляемость процесса.
 - Внедрить систему статусов заказа.
 - Реализовать дашборды и отчеты в реальном времени.
 - Ввести KPI для менеджеров на основе объективных данных из системы.

Для систематизации задач проекта построено дерево целей, декомпозирующее главную миссию на конкретные измеримые подцели. Дерево целей изображено на рисунке 4.

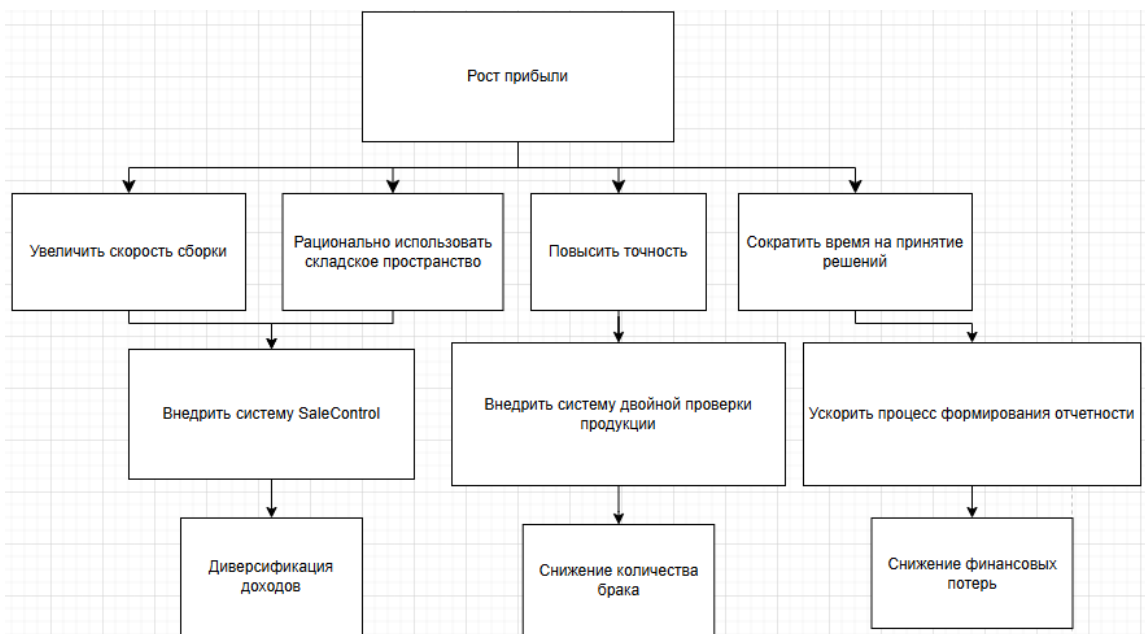


Рисунок 4 – Дерево целей

На схеме представлено иерархическое дерево целей, разработанное для проекта внедрения системы автоматизации управления сбытом готовой продукции (SaleControl). Визуализация выполнена в виде древовидной диаграммы, где вершиной является генеральная стратегическая цель — «Рост прибыли».

Модель «Как должно быть» (To-Be)

С учетом выбранных решений была построена диаграмма процессов будущего состояния. В новой модели внедрена система «SalesControl», которая устраняет выявленные проблемы. Диаграмма изображена на рисунках 5–10.

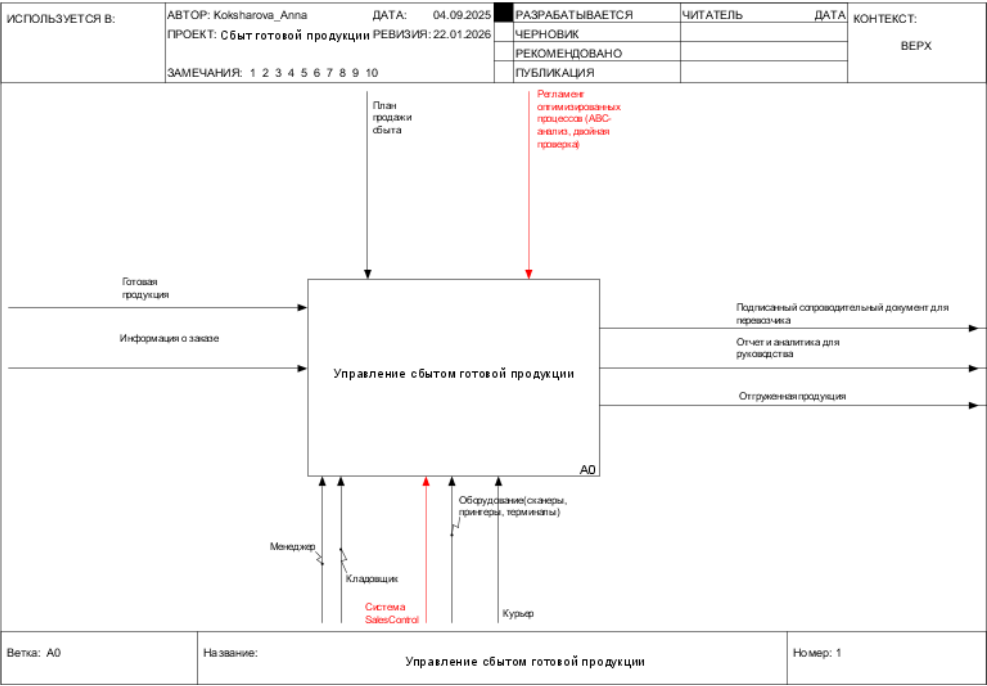


Рисунок 5 – Контекстная диаграмма (A0)

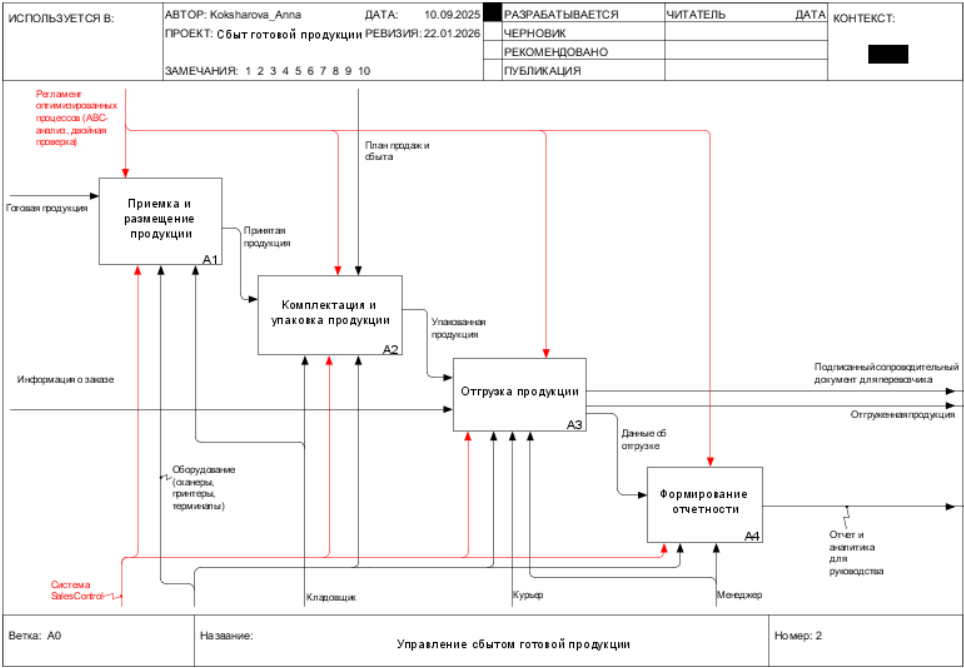


Рисунок 6 – Детализация блока A0

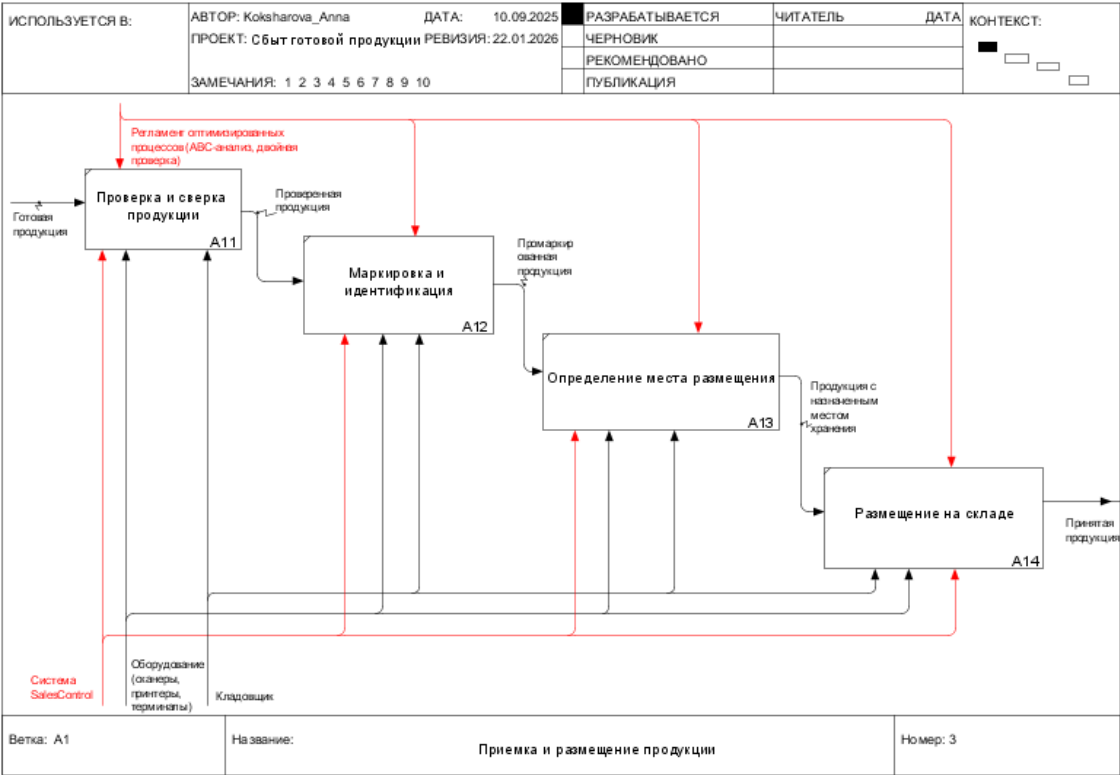


Рисунок 7 – Декомпозиция процесса (A1)

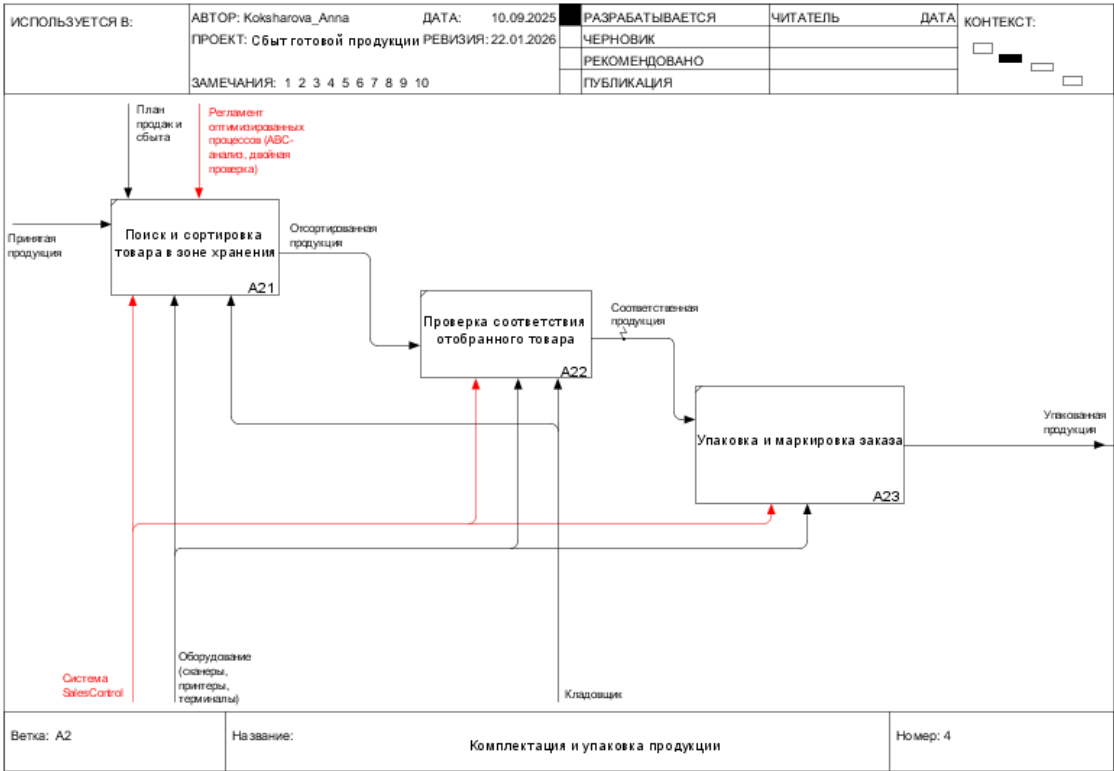


Рисунок 8 – Декомпозиция процесса (A2)

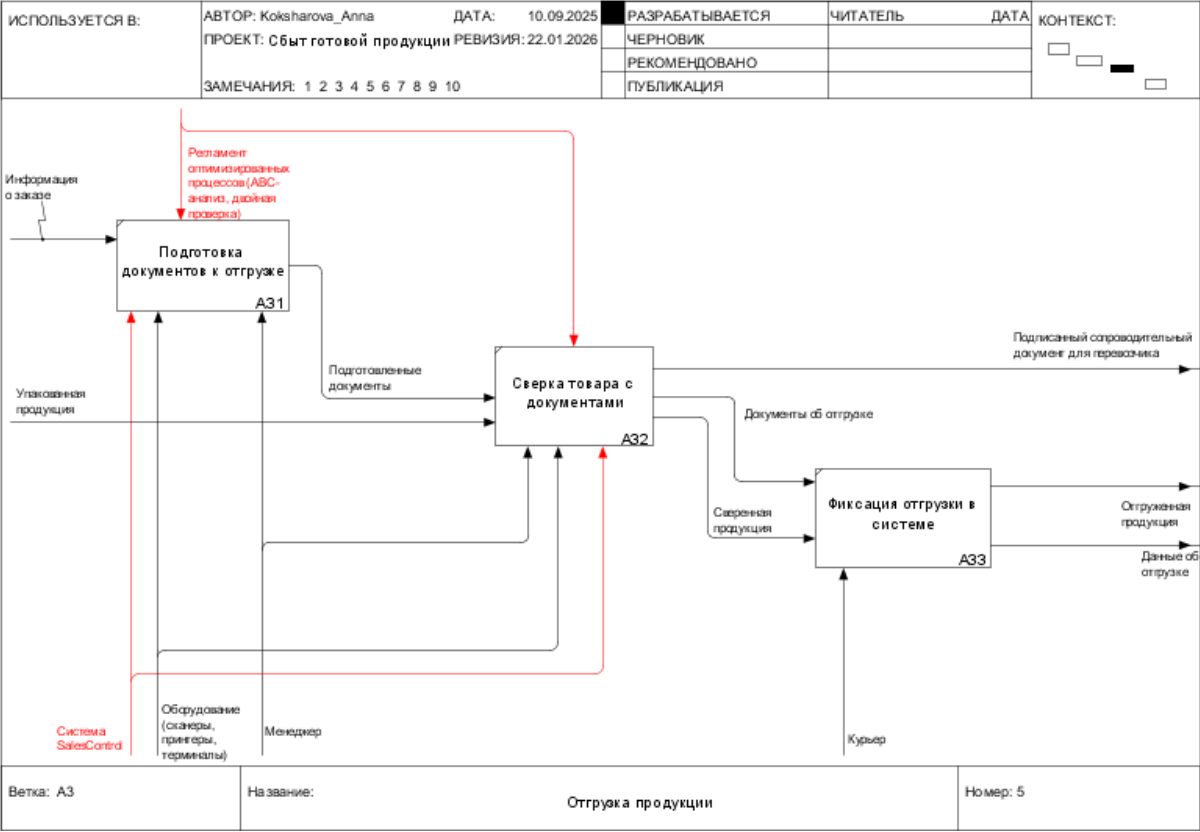


Рисунок 9 – Декомпозиция процесса (A3)

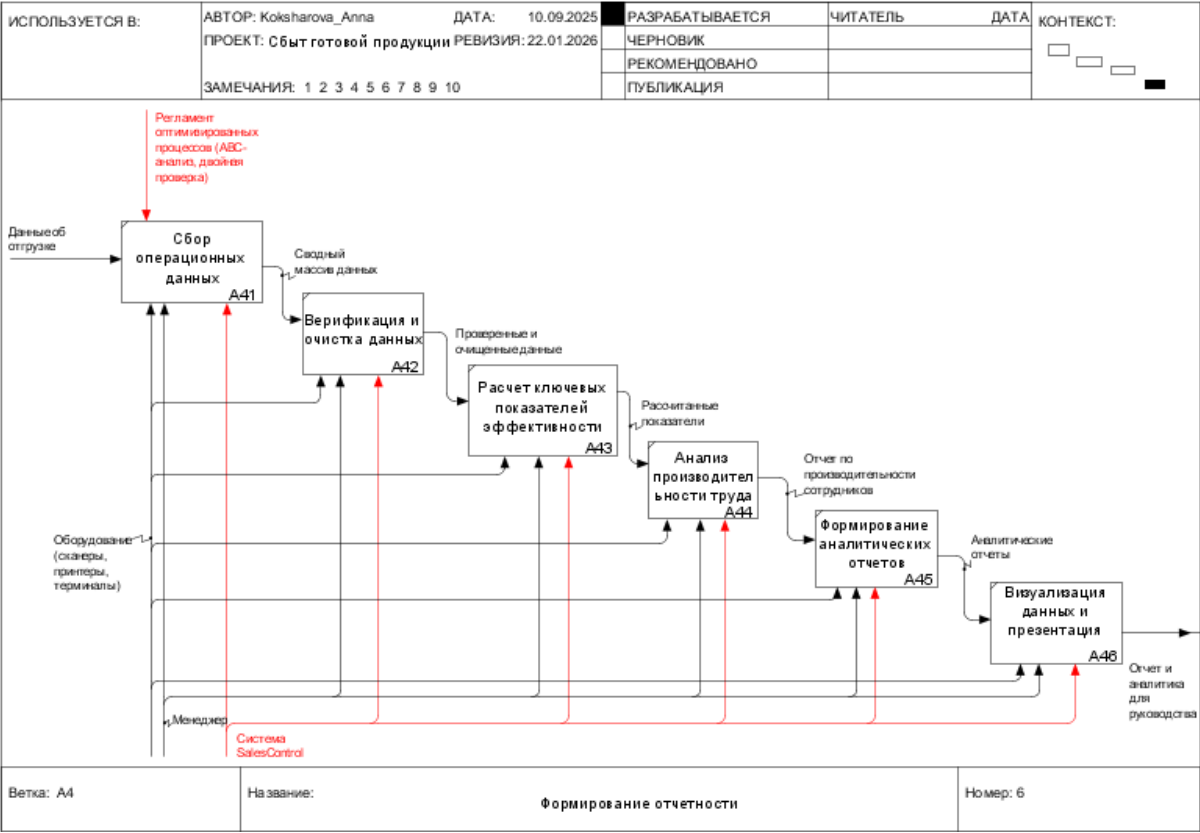


Рисунок 10 – Декомпозиция процесса (A4)

Внедрение системы «SalesControl» кардинально меняет логику работы. Целевой сквозной процесс управления заказом можно описать следующим образом:

1. Создание заказа: менеджер в веб-интерфейсе выбирает клиента из единой базы (или создает нового), добавляет товары из актуального каталога с видимыми остатками. Система автоматически рассчитывает сумму, применяет скидки. При сохранении заказа система автоматически резервирует товары на складе.
2. Обработка на складе: в личном кабинете кладовщика появляется задание на комплектацию с четким списком товаров и мест их хранения. При сканировании или подтверждении отбора система списывает товар с остатков и переводит заказ в статус «Готов к отгрузке».
3. Организация доставки: система автоматически формирует комплект документов (счет-фактура, ТТН). Менеджер или логист назначает на заказ способ доставки (курьер, самовывоз, ТК). Данные для доставки передаются курьеру через его мобильный интерфейс.
4. Исполнение и закрытие: курьер в мобильном приложении фиксирует факт передачи товара клиенту. Система автоматически переводит заказ в статус «Доставлен» и инициирует процесс закрытия (формирование акта, передача данных в бухгалтерию).
5. Аналитика на всех этапах: на каждом шаге данные фиксируются. Руководитель видит на дашборде: количество новых заказов, заказов в работе, среднее время выполнения, выручку, топ менеджеров и товаров.

2.4 Концепции продукта и границы проекта

На основе проведенного анализа сформулирована концепция продукта.

АС «SalesControl» — это корпоративное веб-приложение, предназначенное для автоматизации полного цикла продаж готовой продукции: от приема заказа и резервирования на складе до отгрузки, доставки и формирования аналитической отчетности.

Границы проекта (Минимально жизнеспособный продукт — MVP):
В рамках первой версии системы будут реализованы следующие ключевые функции, направленные на устранение основных «болевых точек»:

1. Управление клиентами (CRM-light): единая база контрагентов с историей заказов.
2. Управление каталогом товаров: ведение номенклатуры с актуальными ценами и остатками.
3. Полный цикл обработки заказа: создание, расчет, онлайн-резервирование, переход по статусам.
4. Складской учет: учет приходов/расходов, задания на комплектацию, инвентаризация.
5. Формирование документов: автоматическая генерация счетов, накладных (ТТН) в PDF.
6. Базовая аналитика и отчеты: дашборд для руководителя, отчеты по продажам и товарам.

Функции, выходящие за рамки MVP (сложная интеграция с 1С для бухгалтерии, мобильное приложение для курьеров с геолокацией, модуль email-рассылок), запланированы в плане развития на последующие этапы.

Полный текст документа «О концепции и границах» приведен в *приложении 1*.

Глава 3. Разработка требований к программной системе

3.1. Определение и документирование пользовательских требований

Пользовательские требования описывают цели, которые пользователи различных ролей должны достигать с помощью системы.

Диаграмма вариантов использования

Для визуального определения границ системы и взаимодействия актеров с её функциями разработана диаграмма вариантов использования (Use Case Diagram) в нотации UML. Для системы «SalesControl» были выделены основные актеры: Менеджер, Курьер, Кладовщик. На рисунке 11 изображена диаграмма вариантов использования «SalesControl».

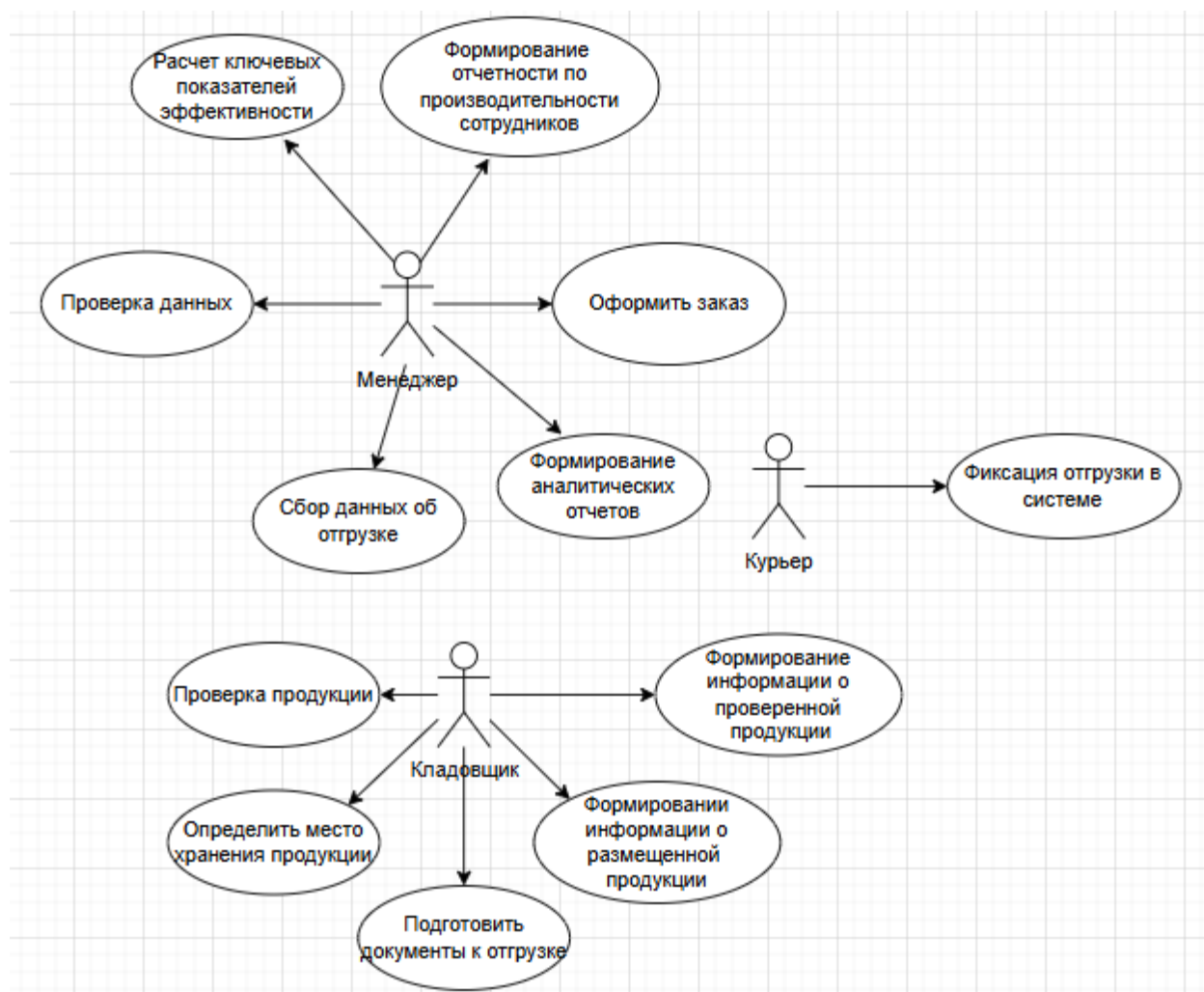


Рисунок 11 – Диаграмма вариантов использования

Диаграмма отражает ключевых актеров (Менеджер, Кладовщик, Курьер) и основные прецеденты их взаимодействия с системой, такие как

«Оформить заказ», «Проверка продукции», «Сформировать отчет по продажам».

Пользовательские истории

Для детализации и приоритизации функциональности диаграмма вариантов использования декомпозирована в набор пользовательских историй, сформулированных по шаблону «Как <Роль>, я хочу <Цель>, чтобы <Польза/Ценность>».

Таблица 3 – Пользовательские истории

Роль	Цель (Что я хочу)	Польза (Зачем мне это нужно)
Менеджер	Я хочу быстро проверить актуальное наличие товара на складе	Чтобы сразу сообщить клиенту о возможности заказа и избежать ситуаций с отсутствующим товаром.
	Я хочу иметь возможность согласовать и зафиксировать в заказе различные условия оплаты (предоплата, отсрочка, оплата по факту)	Чтобы гибко подходить к работе с клиентом и соблюдать финансовую политику компании.
	Я хочу легко оформить заказ на покупку продукции в системе, указав все необходимые данные (товар, количество, цена,	Чтобы заказ был автоматически передан в отдел логистики для дальнейшего выполнения.

	условия оплаты и доставки)	
	Я хочу оперативно предоставлять клиенту информацию о статусе его заказа (подтвержден, собран, передан в доставку)	Чтобы повысить лояльность клиента и снизить количество обращений в поддержку
	Я хочу видеть оформленные заказы, требующие доставки, и иметь возможность назначить на них курьера или транспортную компанию	Чтобы обеспечить своевременную доставку продукции клиенту.
	Я хочу иметь доступ к актуальному каталогу продукции с ценами и описаниями	Чтобы быстро предоставлять полную информацию клиентам и формировать коммерческие предложения.
Кладовщик	Я хочу получать четкие и понятные задания на комплектацию и упаковку товара по каждому заказу	Чтобы быстро и без ошибок подготовить заказ к передаче курьеру или клиенту.
	Я хочу иметь доступ к месту размещения ко всем типам продукции	Чтобы быстро и без ошибок упаковывать заказы.

	Я хочу иметь возможность маркировать продукцию	Чтобы по маркировке быстро определить место хранения продукции
	Я хочу иметь возможность фиксировать отгрузку продукции в системе	Чтобы обновлять данные количества товаров на складе
Курьер	Я хочу иметь возможность фиксировать отгрузку	Чтобы менеджер мог увидеть отгрузку продукции

3.2. Определение и документирование функциональных требований

Функциональные требования описывают внутреннее поведение системы, необходимое для реализации пользовательских запросов.

Диаграмма классов анализа

Для понимания внутренней структуры системы на концептуальном уровне была разработана диаграмма классов анализа. Диаграмма изображена на рисунке 12.



Рисунок 12 – Диаграмма классов анализа системы «SaleControl»

Схема отражает, как пользовательские роли взаимодействуют с функциональными модулями (интерфейсами) и ключевыми информационными объектами системы для выполнения основных бизнес-операций: от оформления и отслеживания отгрузки до складской обработки и формирования аналитики.

Диаграмма последовательности

Диаграммы показывают взаимодействие разных пользователей с системой «SaleControl». Диаграммы последовательности изображены на рисунке 13–15.

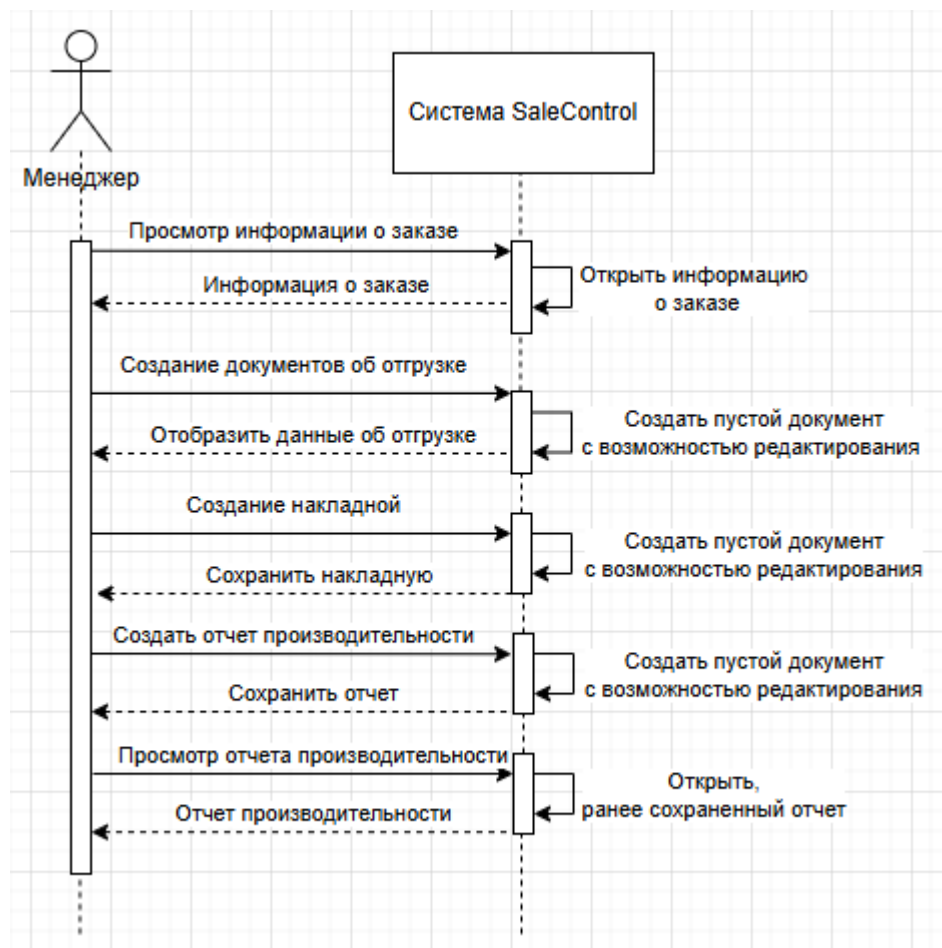


Рисунок 13 – Диаграмма последовательности для менеджера



Рисунок 14 – Диаграмма последовательности для кладовщика

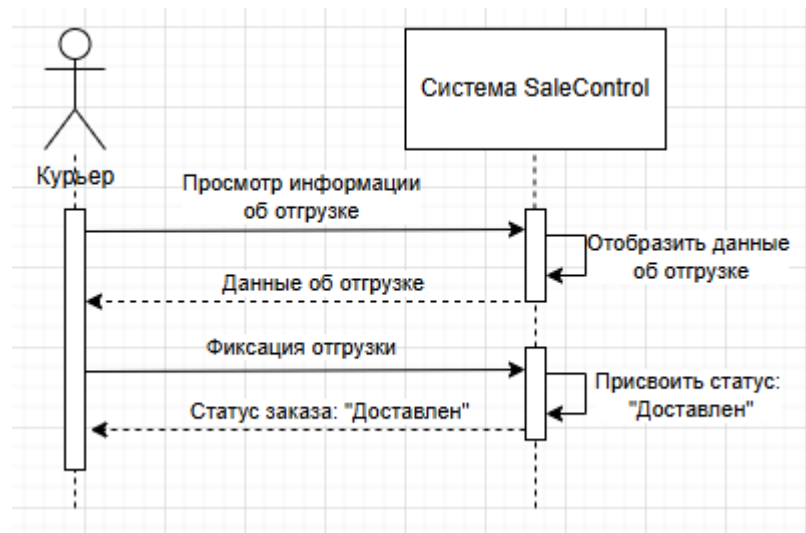


Рисунок 15 – Диаграмма последовательности для курьера

Диаграммы последовательности, моделирующие типичные сценарии взаимодействия Менеджера, Кладовщика и Курьера с системой «SalesControl» для выполнения основных задач.

Диаграмма деятельности

Для описания алгоритмов выполнения сложных бизнес-процессов разработана диаграмма деятельности. Диаграмма деятельности изображена на рисунках 16–18.

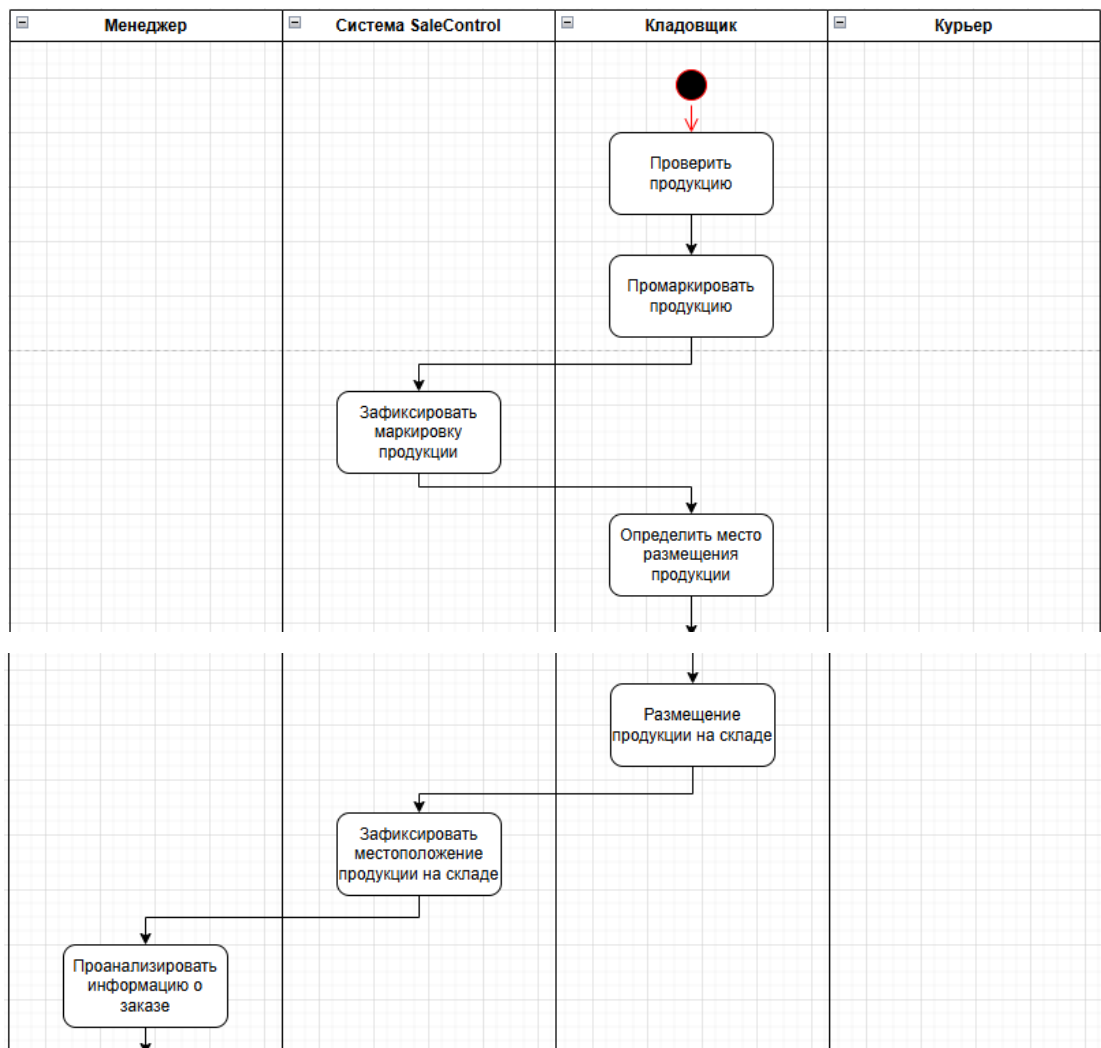


Рисунок 16 – Диаграмма деятельности

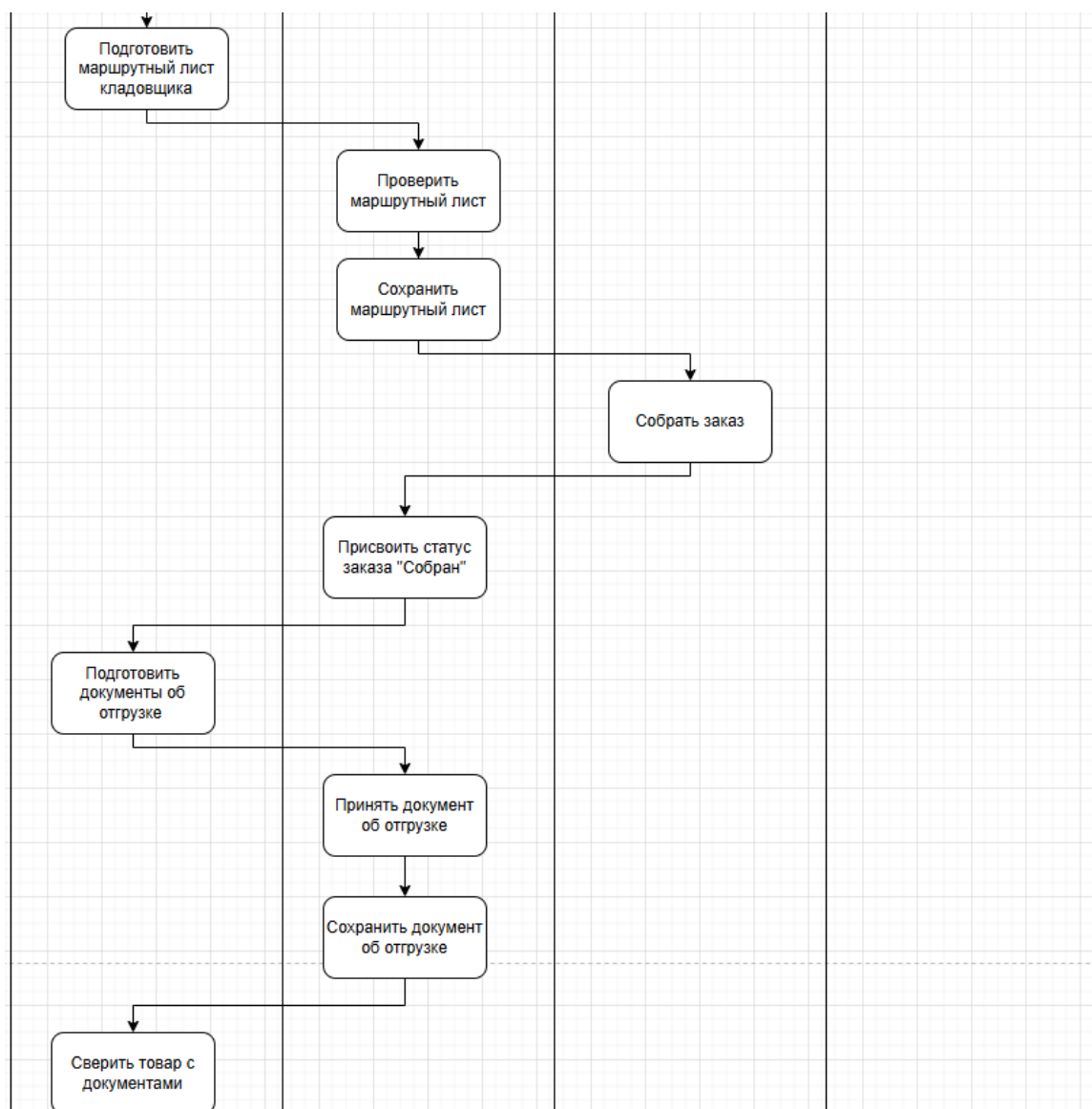


Рисунок 17 – Диаграмма деятельности

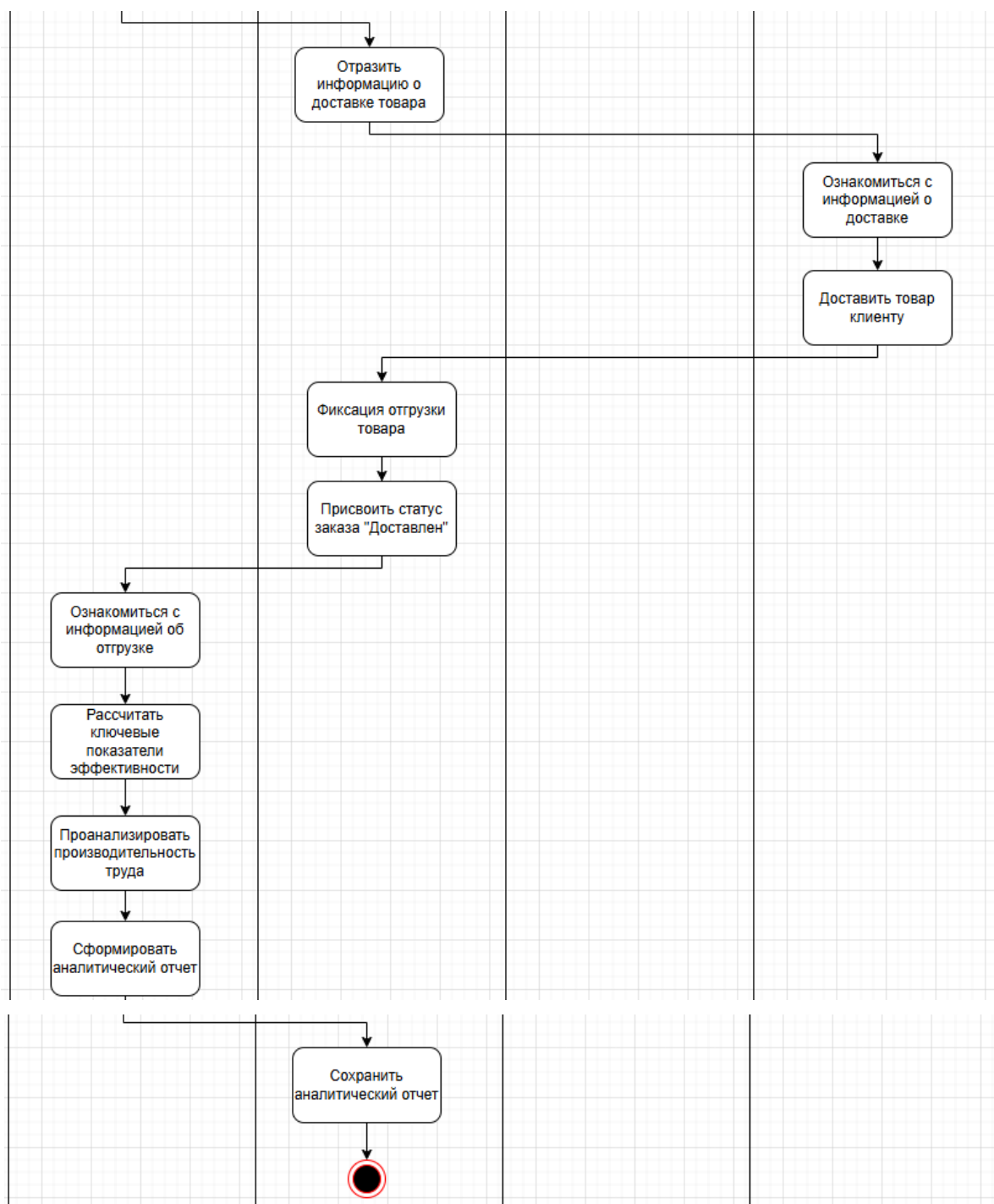


Рисунок 18 - Диаграмма деятельности

Представленная диаграмма деятельности визуализирует бизнес-процесс обработки заказа в системе «SalesControl», синхронизируя действия всех ключевых актеров — Менеджера, Кладовщика и Курьера в едином временном потоке.

Диаграмма классов уровня проектирования

Диаграмма классов служит для наглядного описания структуры будущей системы: какие в ней будут основные сущности (классы), их свойства, методы и взаимосвязи. Диаграмма классов изображена на рисунке 19.

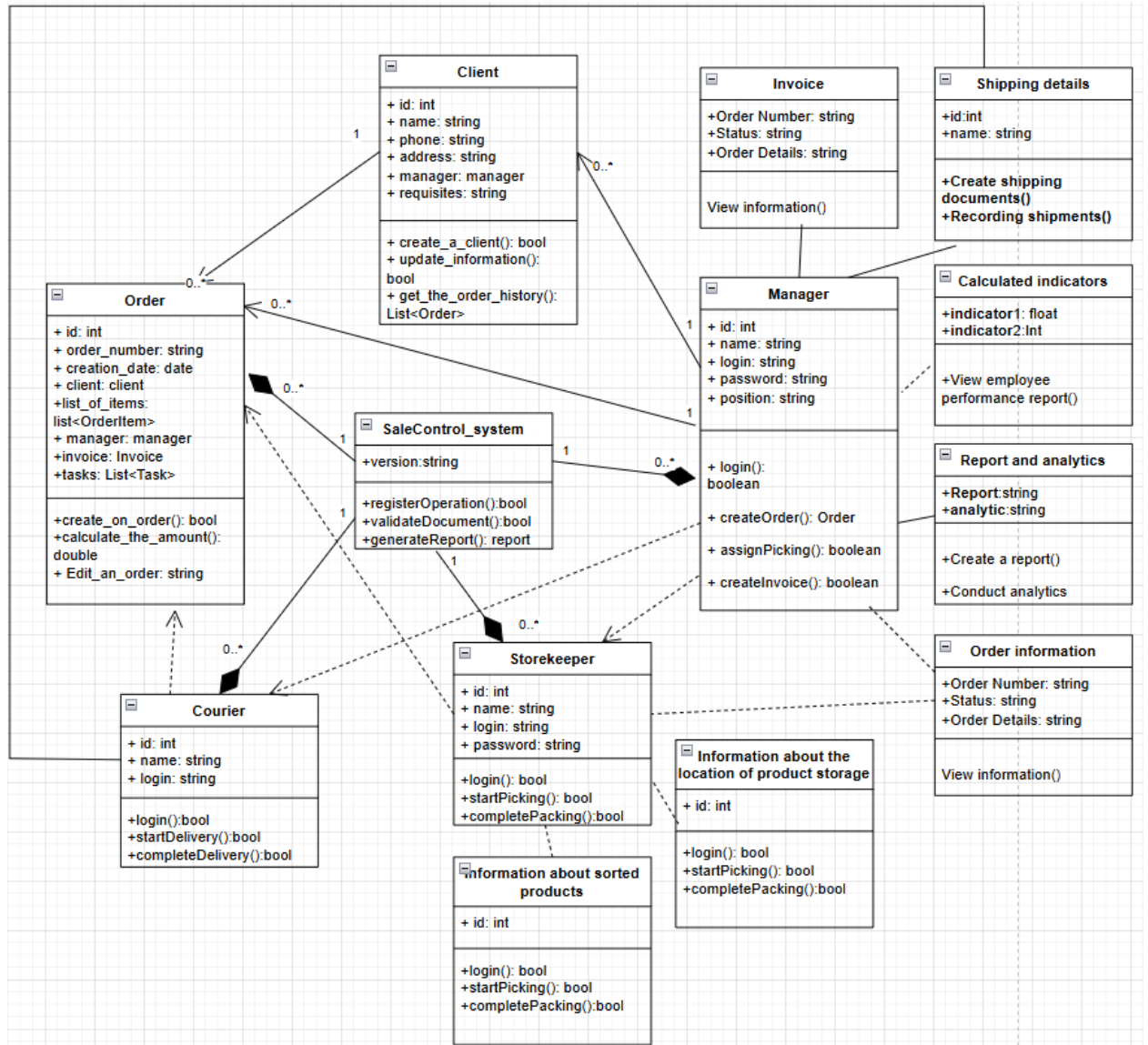


Рисунок 19 – Диаграмма классов

Схема представляет основные бизнес-сущности (Заказ, Клиент, Менеджер, Кладовщик, Курьер, Накладная и т д), их атрибуты и ключевые операции. Модель иллюстрирует тип связей между классами (агрегация, ассоциация), определяя, как данные будут организованы и взаимодействовать в рамках.

Диаграмма «сущность-связь» (ERD)

Для проектирования физической модели базы данных (PostgreSQL) разработана ER-диаграмма. Диаграмма изображена на рисунке 20.

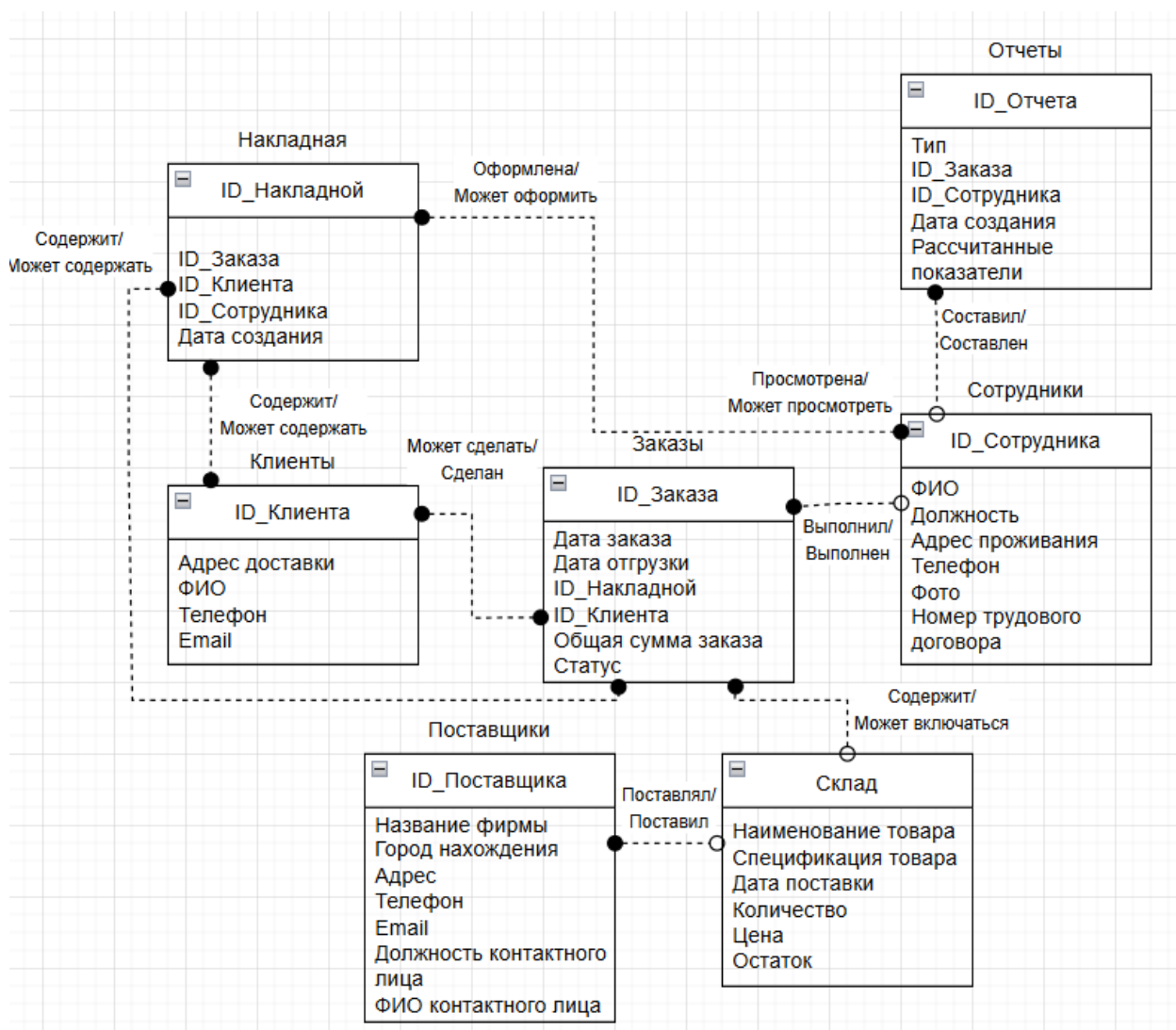


Рисунок 20 – Диаграмма «сущность-связь»

Диаграмма «сущность-связь», представляющая концептуальную модель данных для системы «SalesControl». Схема отображает ключевые сущности предметной области (Накладная, Клиенты, Поставщики, Поставки, Склад) и связи между ними, такие как «Содержит/Может содержать» и «Оформлена/Может оформить».

3.3. Определение и документирование нефункциональных требований

Нефункциональные требования определяют качественные характеристики системы и архитектурные решения.

Пользовательский интерфейс для кладовщика, менеджера и курьера изображен на рисунке 21–22.

The image displays four user interface mockups for a courier system, connected by green arrows indicating navigation flow.

- Авторизация (Authorization):** Features input fields for 'Логин' (Login) and 'Пароль' (Password), and buttons for 'Вход' (Login) and 'Регистрация' (Registration).
- Регистрация (Registration):** Features input fields for 'Логин' (Login), 'Пароль' (Password), and 'Подтвердите пароль' (Confirm password), a 'Зарегистрироваться' (Register) button, and a 'Назад' (Back) button.
- Настройки (Settings):** Includes a 'Размер текста' (Text size) slider, a 'Тема интерфейса' (Interface theme) section with checkboxes for 'Темная' (Dark) and 'Светлая' (Light), and a 'Назад' (Back) button.
- Курьер (Courier) Dashboard:** A main window with two tabs: 'Данные об отгрузках' (Shipment data) and 'Настройки' (Settings). The 'Данные об отгрузках' tab is active, showing a table of shipments with columns for 'Картинка' (Image), 'Номер заказа' (Order number), 'Описание' (Description), 'Статус' (Status), 'Адрес' (Address), and 'Способ оплаты' (Payment method). The 'Настройки' tab shows a 'Штрихкод' (Barcode) input field.

Green arrows indicate the navigation flow: from 'Авторизация' to 'Регистрация', from 'Регистрация' to 'Настройки', from 'Настройки' to 'Курьер', and from 'Курьер' to 'Данные об отгрузках'.

Рисунок 21 – Пользовательский интерфейс

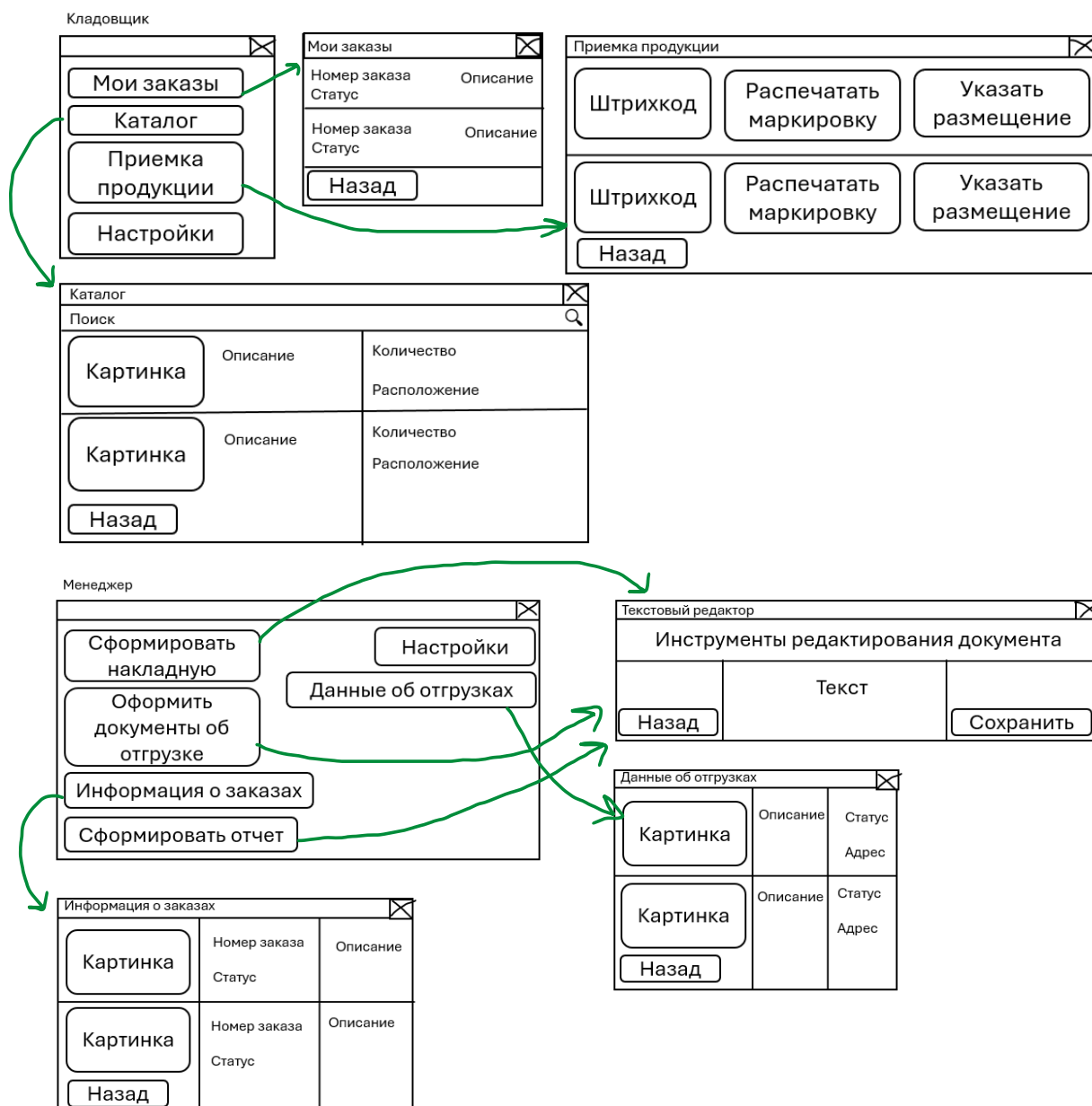


Рисунок 22 – Пользовательский интерфейс

Пользовательские интерфейсы системы «SalesControl» представляют собой целостную экосистему, где каждый актер получает специализированный инструмент для эффективного выполнения своей части сквозного бизнес-процесса — от продажи и складирования до конечной доставки.

Диаграмма компонентов является неотъемлемой частью разработки пользовательского интерфейса. Диаграмма изображена на рисунке 23.

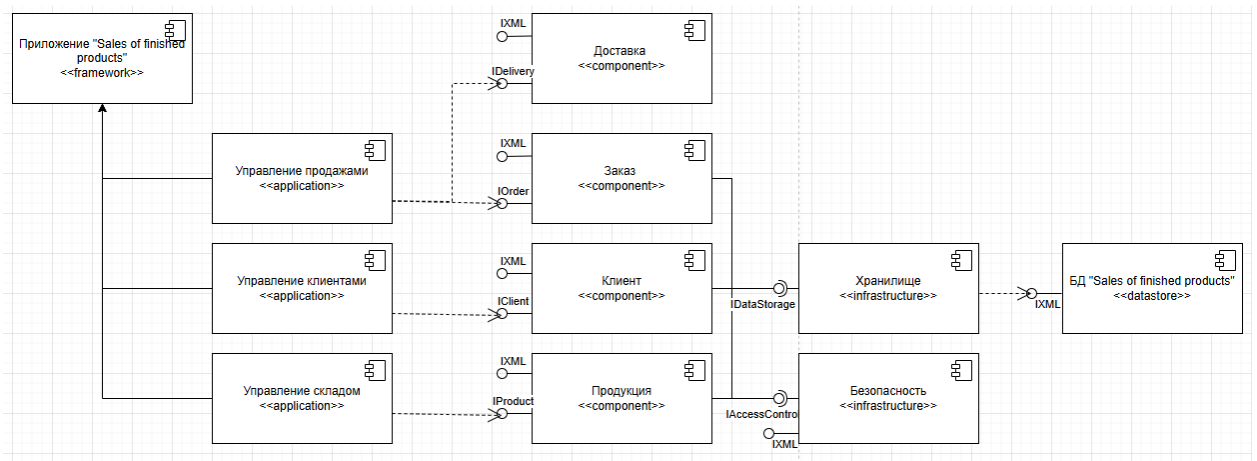


Рисунок 23 – Диаграмма компонентов

На схеме представлены ключевые программные компоненты, их стереотипы (application, component, datastore) и зависимости между ними через интерфейсы (IXML, IDelivery, IProduct).

Диаграмма размещения

Диаграмма размещения описывает физическую расстановку программных узлов на оборудовании. Диаграмма изображена на рисунке 24.

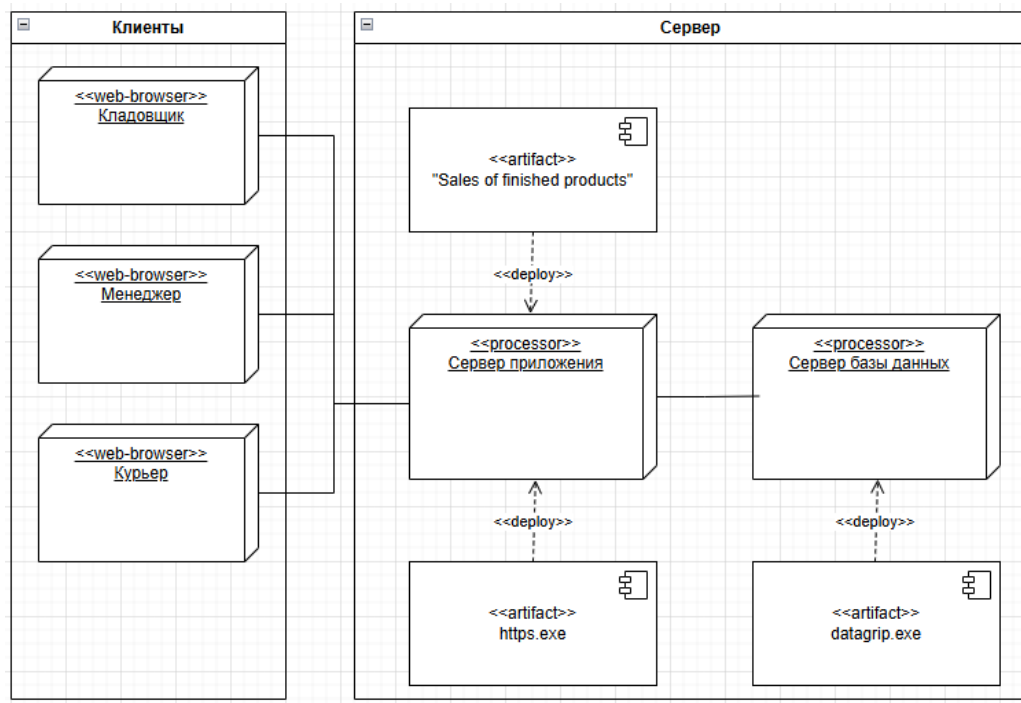


Рисунок 24 – Диаграмма размещения

Диаграмма демонстрирует разделение на клиентскую часть и серверную логику.

Коммуникационные интерфейсы

Для взаимодействия компонентов используются стандартные протоколы:

1. HTTPS/TLS: Шифрование трафика между клиентом и сервером.
2. REST API (JSON): Обмен данными между фронтендом и бэкендом, а также с внешним сервисом парсинга.
3. PostgreSQL Protocol: Связь сервера приложений с базой данных.
4. SMTP: Отправка почтовых уведомлений пользователям.

Полный перечень нефункциональных требований приведен в *приложении 2*.

Заключение

Проведённое исследование и проектная работа позволили обосновать необходимость и сформировать чёткие ориентиры для создания автоматизированной системы управления сбытом готовой продукции «SalesControl».

В ходе предпроектного исследования была получена картина текущего положения дел. Анализ процессов «как есть» выявил системные слабости, характерные для многих производственных компаний: зависимость от разрозненных файлов, постоянные ручные операции, отсутствие оперативного контроля и единого информационного пространства. Эти недостатки имеют далеко идущие последствия, выходящие за рамки простых организационных сложностей, и напрямую ведут к финансовым потерям, ошибкам в обслуживании клиентов и ограничивают потенциал масштабирования бизнеса.

Сравнительный анализ двух состояний — текущего ручного и будущего автоматизированного — наглядно показал качественный скачок, который может быть достигнут: многократное ускорение обработки заказов, практически полное устранение ошибок, связанных с человеческим фактором, и появление реальной, а не декларативной прозрачности всех этапов движения заказа. Это и стало ключевым обоснованием для инициации проекта по созданию собственной системы, идеально адаптированной под уникальные бизнес-процессы компании, в отличие от готовых решений, которые часто требуют болезненной подстройки бизнеса под себя.

Основным результатом работы стала разработка полного комплекса требований, который служит связующим звеном между стратегическими бизнес-целями и технической реализацией. На основе моделирования целевых процессов «как должно быть» была сформулирована чёткая концепция продукта и определён минимально жизнеспособный функционал первой версии системы (MVP). Это позволило расставить приоритеты и сосредоточить усилия на автоматизации самых критичных «болевых точек».

Пользовательские требования, выраженные в виде диаграммы вариантов использования и конкретных историй для менеджера, кладовщика и курьера, зафиксировали, какие именно задачи и с какой пользой должны решать конечные пользователи системы.

Таким образом, цель курсовой работы достигнута. Необходимость создания автоматизированной системы управления сбытом обоснована как экономически, так и технологически. Разработанный комплекс требований представляет собой систематизированный и детализированный проект, который снимает основные неопределённости на старте разработки и минимизирует риски создания продукта, не отвечающего ожиданиям бизнеса и пользователей. Реализация системы «SalesControl» на основе этих материалов откроет возможность перейти от затратного и подверженного ошибкам ручного управления к цифровому, прозрачному и эффективному управлению сбытом, что является обязательным условием конкурентоспособности в современных рыночных условиях.

Список используемых источников

1. Учебное пособие - Ахмедова Х. Г. - Обоснование и разработка требований к программным
2. Учебно-методическое пособие - Ахмедова Х. Г., Овсянникова А. В., Бирюкова А. А. - Обоснование и разработка требований к программным системам ЭУК-конструктор ЭБС Лань
3. Остроух А. В., Суркова Н. Е. - 3.2. Принципы построения модели IDEF0ЭУК-конструктор ЭБС Лань
4. Маран М. М. - Диаграмма классов, стр.33, 3.3. Диаграмма классовЭУК конструктор ЭБС Лань
5. Методические указания по курсовому проектированию – Ахмедова Х. Г., Бирюкова А. А., Овсянникова А. И. - Обоснование и разработка требований к программным системам
6. Дополнительная информация, необходимая для выполнения курсового проекта/работы – Ахмедова Х. Г., Бирюкова А. А., Овсянникова А. И. - Обоснование и разработка требований к программным системам
7. Принципы построения модели IDEF0, использующееся при описании бизнес-процессов организации в состояниях "как есть" и "как должно быть" - А. В. Остроух, Н. Е. Суркова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 164 с.
8. Диаграмма прецедентов (вариантов использования) - Завьялов, А. В. Диаграммы UML для анализа и проектирования информационных систем: учебно-методическое пособие
9. Написание и оформление курсовой работы по дисциплине «Обоснование и разработка требований к программной системе»: методические указания - Х. Г. Ахмедова, А. А. Бирюкова, А. В. Овсянникова.
10. Язык UML. Руководство пользователя - Г. Буч, Дж. Рамбо, А. Джекобсон. – 2-е изд. – М.: ДМК Пресс, 2018. – 496 с.

Приложения

Приложение 1. Документ о концепции и границах

1. Бизнес-требования

1.1 Исходные данные

Организация по сбыту готовой продукции использует неэффективные, примитивные инструменты для управления складскими и логистическими процессами. Основными инструментами являются Excel, Word и бумажный документооборот. Это приводит к полному отсутствию автоматизации, что является ключевой проблемой, требующей решения.

1.2 Возможности бизнеса

Внедрение современной системы управления складом (WMS) откроет для бизнеса следующие возможности:

- Ликвидация ошибок учета: Автоматизация данных исключит опечатки, ошибки в подсчетах и путаницу в отгрузках.
- Повышение операционной эффективности: кратное увеличение скорости обработки заказов, формирования документов и поиска товара на складе.
- Полный контроль над логистической цепочкой: отслеживание каждой товарной единицы от приемки до отгрузки.
- Снижение операционных затрат: потенциал для сокращения затрат на 40% в первый год за счет оптимизации процессов.
- Улучшение клиентского сервиса: возможность в режиме реального времени предоставлять клиентам информацию об остатках и статусе их заказов.

1.3 Бизнес-цели

1. Повышение точности учета: достичь почти 100% точности данных об остатках и перемещениях товара.
2. Сокращение времени обработки заказов: уменьшить время от поступления заявки до отгрузки как минимум в 2 раза.

3. Оптимизация использования ресурсов: повысить производительность труда сотрудников и эффективность использования складских площадей.
4. Минимизация потерь: свести к нулю потери от краж, неучтенных списаний и брака.
5. Обеспечение прозрачности: предоставить руководству инструменты для реального контроля и анализа всех складских операций.

1.4 Критерии успеха

- Количественные:
 - Сокращение операционных затрат на 40% в первый год.
 - Уменьшение количества ошибок при комплектации заказов до менее 0.5%.
 - Сокращение времени на формирование документов (накладных, ТТН) с часов до минут.
 - Увеличение оборачиваемости товара на X%.
- Качественные:
 - Возможность в любой момент ответить на вопрос о точных остатках и месте нахождения любого товара.
 - Полное устранение путаницы в отгрузках.
 - Повышение удовлетворенности клиентов скоростью и точностью обслуживания.

1.5 Положение о концепции проекта

Для сотрудников склада, которым необходимо эффективно управлять товарными потоками, данная система управления складом (SaleControl) является ключевым инструментом, который обеспечит полный контроль над приемкой, хранением, комплектацией и отгрузкой продукции. Система будет отслеживать местоположение каждой товарной единицы, фиксировать точное количество товара на остатках и вести полную историю перемещений. В отличие от текущей ручной системы, наш продукт автоматизирует формирование отчетов и документооборот, что позволит сократить

операционные затраты на 40% в первый год эксплуатации за счет снижения ошибок, оптимизации площадей и минимизации потерь.

1.6 Бизнес-риски

1. Сопротивление персонала: сотрудники могут сопротивляться внедрению новых технологий из-за непонимания или страха перед изменениями.
2. Недостаточное обучение: некачественное обучение пользователей приведет к ошибкам и неполному использованию функционала системы.
3. Сбои в работе системы: технические сбои или ошибки в программном коде могут парализовать работу склада на время их устранения.
4. Неточность данных на старте: неверное или неполное внесение начальных остатков и данных о номенклатуре приведет к некорректной работе системы.
5. Превышение бюджета или сроков: сложность интеграции или доработок может привести к задержкам в реализации и увеличению стоимости проекта.

1.7 Предположения и зависимости

- Предположения:
 1. У сотрудников есть базовые навыки работы с мобильными устройствами.
 2. На складе существует или может быть внедрена система штрихкодирования для всей номенклатуры.
 3. Руководство готово выделить необходимые ресурсы на закупку оборудования (сканеры, терминалы, принтеры этикеток).
- Зависимости:
 1. Доступность и стабильность работы Wi-Fi сети на всей территории склада.
 2. Наличие технического специалиста (или подрядчика) для поддержки и администрирования системы.

2. Рамки и ограничения проекта

2.1 Основные функции

- Управление пользователями и доступом: ролевая модель (Кладовщик, Менеджер и Курьер).
- Приемка и размещение: сканирование штрихкодов, фиксация местоположения, печать этикеток.
- Комплектация заказов: пошаговая инструкция для кладовщика с проверкой через сканирование.
- Отгрузка: сканирование и списание отгруженного товара, автоматическое формирование документов.
- Отслеживание доставки: функционал для курьера по подтверждению приема и сдачи груза.
- Отчетность и аналитика: дашборды для руководителя с KPI (остатки, оборачиваемость, производительность).

2.2 Объем первоначально запланированной версии (MVP - Минимально жизнеспособный продукт)

Первая версия системы должна включать критически важный функционал для автоматизации ключевых процессов:

- Аутентификация и ролевая модель.
- Приемка продукции со сканированием и размещением.
- Комплектация заказов кладовщиком с обязательным сканированием.
- Отгрузка со сканированием и печатью.
- Базовый отчет по текущим остаткам.
- Функционал для водителя по сканированию при приеме и сдаче груза.

2.3 Объем последующих версий

- Версия 2.0: Расширенная аналитика и дашборды KPI для руководителя, проведение инвентаризации через систему, интеграция с 1С или бухгалтерскими системами.

- Версия 3.0: Оптимизация маршрутов комплектации (wave picking), управление складскими зонами (стеллажи, ячейки), расширенный модуль управления транспортом и пулом перевозчиков.

2.4 Ограничения и исключения

- Ограничения:
 - Система не управляет автономно складской техникой (роботы-паллетотеры, конвейеры).
 - Для работы требуется стабильное интернет-соединение.
- Исключения:
 - В рамках данного проекта не разрабатывается система управления взаимоотношениями с клиентами (CRM).
 - Не предусмотрена автоматическая закупка товаров или управление заказами поставщикам.

3. Бизнес-контекст

3.1 Профили заинтересованных лиц

- Кладовщик: заинтересован в простом и понятном интерфейсе для быстрой приемки и точного размещения товара. Критически важна работа на мобильном устройстве.
- Менеджер: Заинтересован в едином центре контроля, прозрачности всех процессов и оперативном доступе к данным об остатках и статусах заказов. Также в снижении издержек, повышении оборачиваемости и полном контроле над материальными ценностями.
- Курьер: Заинтересован в четком плане задач и простом подтверждении операций приема-передачи груза.

3.2 Приоритеты проекта

1. Высший приоритет: Автоматизация учета остатков и их местоположения (ликвидация "слепых зон").
2. Высокий приоритет: Автоматизация процессов комплектации и отгрузки для исключения ошибок.

3. Средний приоритет: Внедрение мобильного интерфейса для сотрудников на складе.
4. Низкий приоритет: Разработка расширенных аналитических отчетов и дашбордов.

3.3 Особенности развертывания

- Пилотное внедрение: Внедрение системы будет поэтапным. Начнется с одного склада или одной смены для отладки процессов и обучения персонала.
- Обучение: Обязательное проведение полноценного обучения для всех категорий пользователей с написанием инструкций.
- Миграция данных: Необходима тщательная процедура переноса текущих остатков и номенклатуры из существующих систем (Excel) в новую SalesControl с последующей сверкой.

Приложение 2. Атрибуты качества

Внешние атрибуты качества:

1. Доступность (Availability): Степень готовности системы к работе в требуемый момент времени.
 - Пример требования: AVL-1. Система должна быть доступна для пользователей 99.5% времени в рабочее время (с 09:00 до 18:00 по местному времени).
2. Удобство установки (Installability): Легкость, с которой систему можно установить и деинсталлировать в целевой среде.
 - Пример требования: INS-1. Установка клиентской части приложения на стандартной конфигурации Windows должна занимать не более 5 минут и не требовать от пользователя ручного редактирования файлов конфигурации.
3. Целостность (Integrity): Способность системы защищать данные от несанкционированного доступа, модификации или разрушения.
 - Пример требования: INT-1. Система должна гарантировать, что финансовые транзакции либо выполняются полностью, либо не выполняются вовсе (принцип атомарности).
4. Совместимость (Interoperability): Способность системы взаимодействовать с другими системами и обмениваться с ними информацией.
 - Пример требования: IOP-1. Система должна импортировать данные о клиентах из CRM-системы "1C" через стандартный API в формате JSON.
5. Производительность (Performance): Скорость и эффективность реакции системы на действия пользователя или другие системы.
 - Пример требования: PER-1. 95% всех поисковых запросов в каталоге должны обрабатываться менее чем за 2 секунды при одновременной работе до 1000 пользователей.
6. Надёжность (Reliability): Способность системы выполнять требуемые

функции при заданных условиях в течение заданного периода времени без сбоев.

- Пример требования: REL-1. Среднее время наработки на отказ (MTBF) для критических модулей системы должно составлять не менее 10 000 часов.

7. Устойчивость (Robustness): Способность системы корректно функционировать при некорректных входных данных, перегрузках или сбоях в окружающей среде.

- Пример требования: ROB-1. При получении некорректных данных от внешнего сервиса система должна занести данные в очередь на повторную обработку и уведомить администратора, не прекращая общую работу.

8. Защита (Safety): Способность системы не создавать опасных ситуаций для людей, оборудования или окружающей среды.

- Пример требования: SAF-1. При обнаружении критического перегрева двигателя система управления должна автоматически перевести его в безопасный режим и подать звуковой сигнал оператору.

9. Безопасность (Security): Способность системы защищать информацию и данные от несанкционированного доступа, раскрытия или модификации.

- Пример требования: SEC-1. Все пароли пользователей должны храниться в базе данных только в хэшированном виде с использованием стойкого алгоритма (например, bcrypt).

10. Удобство использования (Usability): Легкость, с которой пользователи могут освоить, использовать систему и добиться с ее помощью поставленных целей.

- Пример требования: USE-1. 90% новых пользователей должны успешно завершить процедуру регистрации и создать свой первый проект без обращения в службу поддержки.

Внутренние атрибуты качества:

1. Эффективность (Efficiency): Степень использования системой вычислительных ресурсов (процессорного времени, памяти, пропускной способности сети).
 - Пример требования: EFF-1. В режиме простоя серверное приложение не должно использовать более 1% ресурсов CPU и 500 МБ оперативной памяти.
2. Возможность модификации (Modifiability): Легкость, с которой систему можно изменять для исправления ошибок, улучшения производительности или адаптации к изменяющимся требованиям.
 - Пример требования: MOD-1. Добавление нового типа платежной системы (например, Crypto) не должно требовать изменения кода ядра платежного модуля, а только реализации отдельного плагина.
3. Переносимость (Portability): Легкость, с которой систему можно перенести из одной операционной среды в другую (аппаратную платформу, операционную систему, базу данных).
 - Пример требования: POR-1. Система должна быть способна работать как на СУБД PostgreSQL, так и на Microsoft SQL Server без изменения исходного кода, только путем изменения строки подключения.
4. Возможность повторного использования (Reusability): Степень, в которой компоненты системы могут быть использованы в других программных системах.
 - Пример требования: REU-1. Модуль аутентификации и авторизации должен быть реализован в виде отдельной библиотеки, которую можно подключить к другим внутренним проектам компании.
5. Масштабируемость (Scalability): Способность системы увеличивать свою производительность при добавлении ресурсов (аппаратных или программных).

- Пример требования: SCA-1. Система должна поддерживать горизонтальное масштабирование: добавление нового сервера приложений в кластер должно увеличивать общую пропускную способность на 80–90%.
6. Проверяемость (Verifiability): Легкость, с которой можно проверить, соответствует ли система требованиям (например, путем тестирования, анализа кода).
- Пример требования: VER-1. Каждый программный модуль должен иметь модульные тесты, обеспечивающие покрытие кода не менее 85%.