**СОДЕРЖАНИЕ**

[1. ВВЕДЕНИЕ 3](#_Toc205730455)

[1.1 Наименование разработки 3](#_Toc205730456)

[1.2 Основание для разработки 3](#_Toc205730457)

[1.3 Источники разработки 3](#_Toc205730458)

[1.4 Назначение и цели создания системы 5](#_Toc205730459)

[1.5 Область применения 5](#_Toc205730460)

[1.6 Краткое описание объекта автоматизации 6](#_Toc205730461)

[1.7 Уровень документа 6](#_Toc205730462)

[2. НАЗНАЧЕНИЕ РАЗРАБОТКИ 6](#_Toc205730463)

[2.1 Цель разработки системы 6](#_Toc205730464)

[2.2 Основные задачи системы 7](#_Toc205730465)

[2.3 Основные пользователи системы 8](#_Toc205730466)

[2.4 Бизнес-процессы, подлежащие автоматизации 8](#_Toc205730467)

[2.5 Ожидаемые результаты внедрения 8](#_Toc205730468)

[2.6 Обоснование необходимости разработки 9](#_Toc205730469)

[3. ТРЕБОВАНИЯ К ФУНКЦИОНАЛЬНЫМ ХАРАКТЕРИСТИКАМ 9](#_Toc205730470)

[3.1 Общие положения 9](#_Toc205730471)

[3.2 Функциональные требования 10](#_Toc205730472)

[3.3 Бизнес-правила 12](#_Toc205730473)

[3.4 Потоки данных 13](#_Toc205730474)

[3.4.1 Поток: Создание новой ТТ 13](#_Toc205730475)

[3.4.2 Поток: Перепривязка ТТ 14](#_Toc205730476)

[3.4.3 Поток: Сверка накладных 14](#_Toc205730477)

[3.5 Структура сущности `TradePoint` 15](#_Toc205730478)

[3.6 Триггеры и реакции системы 16](#_Toc205730479)

[4. ТРЕБОВАНИЯ К НАДЁЖНОСТИ 16](#_Toc205730480)

[4.1 Общие положения 16](#_Toc205730481)

[4.2 Отказоустойчивость 17](#_Toc205730482)

[4.3 Сохранность данных 17](#_Toc205730483)

[4.4 Восстановление после сбоев 18](#_Toc205730484)

[4.5 Обработка ошибок 19](#_Toc205730485)

[4.6 Бизнес-правила надёжности 19](#_Toc205730486)

[4.7 Схема обработки сбоев 20](#_Toc205730487)

[5. УСЛОВИЯ ЭКСПЛУАТАЦИИ 21](#_Toc205730488)

[5.1 Общие положения 21](#_Toc205730489)

[5.2 Требования к аппаратному обеспечению 21](#_Toc205730490)

[5.3 Требования к программному обеспечению 22](#_Toc205730491)

[5.4 Сетевые требования 22](#_Toc205730492)

[5.5 Ограничения по использованию 22](#_Toc205730493)

[6. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 23](#_Toc205730494)

[6.1 Общие положения 23](#_Toc205730495)

[6.2 Перечень программной документации 24](#_Toc205730496)

[6.3 Требования к содержанию документации 24](#_Toc205730497)

[6.3.1 Руководство пользователя 24](#_Toc205730498)

[6.3.2 Описание программы 25](#_Toc205730499)

[6.3.3 Описание применения 25](#_Toc205730500)

[6.3.5 Программа и методика испытаний 26](#_Toc205730501)

[6.4 Форматы и средства оформления 26](#_Toc205730502)

[6.5 Требования к актуализации документации 27](#_Toc205730503)

[7. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ 27](#_Toc205730504)

[7.1 Общие положения 27](#_Toc205730505)

[7.2 Программные компоненты системы 28](#_Toc205730506)

[7.3 Внешние системы и точки интеграции 28](#_Toc205730507)

[7.4 Модель хранения настроек 29](#_Toc205730508)

[7.5 Пути к файлам 30](#_Toc205730509)

[7.6 Инструменты разработки и сопровождения 31](#_Toc205730510)

[7.7 Управление настройками через GUI 31](#_Toc205730511)

[7.8 Требования к внешним системам 31](#_Toc205730512)

[7.9 Архитектура интеграции 32](#_Toc205730513)

[8. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ 33](#_Toc205730514)

[8.1 Общие положения 33](#_Toc205730515)

[8.2 Источники данных 33](#_Toc205730516)

[8.3 Структура базы данных 34](#_Toc205730517)

[8.3.1 Таблица `trade\_points` 34](#_Toc205730518)

[8.3.2 Таблица `region\_esr\_mapping` 35](#_Toc205730519)

[8.3.3 Таблица `invoice\_reconciliation` 35](#_Toc205730520)

[8.4 Форматы входных и выходных данных 36](#_Toc205730521)

[8.4.1 Входные данные 36](#_Toc205730522)

[8.4.2 Выходные данные 39](#_Toc205730523)

[8.5 Правила обработки данных 39](#_Toc205730524)

[8.6 Схема синхронизации данных 41](#_Toc205730525)

[8.7 Требования к сохранности данных 42](#_Toc205730526)

[9. МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ 42](#_Toc205730527)

[9.1 Общие положения 42](#_Toc205730528)

[9.2 Угрозы безопасности 43](#_Toc205730529)

[9.3 Защита конфиденциальных данных 43](#_Toc205730530)

[9.4 Средства защиты 44](#_Toc205730531)

[9.5 Ограничения системы безопасности 45](#_Toc205730532)

[9.6 Рекомендации по безопасной эксплуатации 45](#_Toc205730533)

[9.7 Обработка инцидентов 46](#_Toc205730534)

[9.8 Требования к защите при хранении и передаче 46](#_Toc205730535)

[10. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 47](#_Toc205730536)

[10.1 Общие положения 47](#_Toc205730537)

[10.2 Стадии разработки 47](#_Toc205730538)

[10.3 План спринтов 48](#_Toc205730539)

[10.4 Ответственные роли 51](#_Toc205730540)

[10.5 Критерии приёма системы 51](#_Toc205730541)

[10.6 Порядок контроля и приёмки 52](#_Toc205730542)

[10.7 Выходные документы по стадиям 52](#_Toc205730543)

1. ВВЕДЕНИЕ

1.1 Наименование разработки

Система автоматизации создания торговых точек и сверки накладных с использованием AI-агента (далее — Система)

1.2 Основание для разработки

Разработка осуществляется на основании:

- Необходимости автоматизации рутинных процессов: создания торговых точек (ТТ) в CRM и сверки накладных с DMS

- Потребности в снижении трудозатрат, минимизации ошибок и ускорении обработки запросов от менеджеров

1.3 Источники разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Документ / Источник | Описание | Формат | Автор / Ответственный |
| 1 | Product Backlog | Полный список user story и задач, согласованный с заказчиком | XLSX, product\_backlog.xlsx | Аналитик, ИИ-ассистент |
| 2 | Таблица атрибутов торговых точек | Excel-файл с историей созданных ТТ: названия, адреса, ESR, направления деятельности и др. | Excel | Аналитик |
| 3 | Справочник районов ТП ESR | Таблица соответствия кода района маршрута (например, Н02) и кода ESR | Excel, 1С (справочник "Районы ТП") | Аналитик, 1С |
| 4 | Описание процессов в 1С | Описание действий аналитика при поиске ТТ, проверке XCRM GUID, извлечении данных | Внутренняя документация | Аналитик |
| 5 | Техническая архитектура системы | Архитектура: operation-process-view, модули Python, интеграция с Selenium, PostgreSQL, LLM | Схема, описанная в диалоге | Аналитик, ИИ-ассистент |
| 6 | Примеры запросов от менеджеров | Типовые запросы от менеджеров | Текстовые сообщения | Менеджеры по продажам |

1.4 Назначение и цели создания системы

Цель разработки — создание программного обеспечения, позволяющего автоматизировать:

- Создание и пере привязку торговых точек в CRM по запросам менеджеров

- Сверку накладных с системой DMS

- Поддержку ручного и автоматического ввода данных через GUI и AI-чат

- Интеграцию с внешними системами: CRM, DMS, Excel, PostgreSQL

Основные цели:

- Сокращение времени на обработку одного запроса с 1-2 минут до 30 секунд и меньше

- Устранение ручных ошибок при заполнении форм

- Обеспечение прозрачности процессов (логирование, отчёты)

- Поддержка масштабирования: от ручного ввода к автономной работе через чат-бот

1.5 Область применения

Система предназначена для использования:

- Аналитиками отдела продаж

- Менеджерами по работе с партнёрами

- IT-специалистами (для мониторинга и поддержки)

Сфера применения:

- Внутренние процессы компании по интеграции с внешними торговыми партнёрами

- Автоматизация взаимодействия с CRM и DMS через веб-интерфейсы

- Поддержка принятия решений на основе исторических данных и AI-анализа

1.6 Краткое описание объекта автоматизации

Тип системы - Десктопное приложение с GUI и AI-агентом

Язык программирования - Python 3.13+

Фреймворки - PyQt5, Selenium, pandas, openpyxl, xlwings

База данных - PostgreSQL 14+

Искусственный интеллект - Интеграция с LLM через API open router

Архитектура - operation-process-view с модульной структурой

Режим работы - Интерактивный (GUI + чат), фоновые процессы

1.7 Уровень документа

Настоящее ТЗ соответствует требованиям ГОСТ 34.602-89 "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы".

2. НАЗНАЧЕНИЕ РАЗРАБОТКИ

2.1 Цель разработки системы

Целью разработки является создание масштабируемого, тестируемого и документированного программного обеспечения, предназначенного для автоматизации рутинных операций аналитика отдела продаж.

Конечная цель — не просто автоматизация, а построение устойчивой, поддерживаемой и расширяемой системы, которая:

* Имеет чёткую архитектуру (operation-process-view)
* Поддерживает модульность и повторное использование кода
* Сопровождается полной технической и пользовательской документацией
* Включает инструкции по установке, настройке, использованию и расширению
* Обеспечивает прозрачность процессов через логирование, уведомления и отчёты
* Поддерживает автоматизированное тестирование ключевых сценариев
* Готова к интеграции с внешними системами (1С, CRM, DMS) на всех уровнях

2.2 Основные задачи системы

|  |  |  |
| --- | --- | --- |
| № | Задача | Описание |
| 1 | Автоматизация создания ТТ в CRM | На основе запроса от менеджера и уточнённых данных из 1С система формирует и заполняет форму в CRM через Selenium |
| 2 | Пере привязка ТТ к новому ESR | По XCRM GUID система находит ТТ, удаляет старые ESR-привязки, добавляет новые, не затрагивая MRC |
| 3 | Сверка накладных с DMS | Загрузка Excel-файла → фильтрация по SFA пусто ISA ≠ 0 → проверка наличия в DMS → визуализация результата |
| 4 | Поддержка AI-агента | Чат-интерфейс для приёма запросов, извлечения данных, уточнения недостающей информации, генерации JSON |
| 5 | Фоновая обработка | Все процессы выполняются в фоне, без блокировки GUI, если соответствующая настройка включена |
| 6 | Логирование и уведомления | Все операции и ошибки логируются в файл и БД, пользователь получает уведомление о результате |
| 7 | Хранение и анализ данных | Данные о ТТ и маппинг район → ESR хранятся в PostgreSQL для поиска, фильтрации и обучения AI |
| 8 | Экспорт результатов | Результаты сохраняются в Excel с цветовой заливкой и текстовыми пометками для передачи в дальнейший анализ ошибок |

2.3 Основные пользователи системы

Основными пользователями системы являются аналитики отдела продаж, но также данное приложение может быть предоставлено к пользованию или интегрировано в другие системы для прямого создания торговых точек менеджерами по запросу к AI агенту. Дальнейшее расширение и интеграция происходят по согласованию с руководителем аналитиков отдела продаж, руководителем менеджеров и командой разработки приложения.

2.4 Бизнес-процессы, подлежащие автоматизации

|  |  |  |
| --- | --- | --- |
| Процесс | Текущее состояние | Цель автоматизации |
| Создание ТТ в CRM | Ручной ввод в CRM по данным из 1С | Автоматизация через Selenium на основе данных из GUI или чата с AI |
| Перепривязка ТТ к ESR | Поиск по GUID → редактирование привязок вручную | Автоматическое удаление ESR-привязок, добавление новых |
| Сверка накладных | Ручной поиск в DMS по списку из Excel | Автоматическая проверка всех накладных из отфильтрованного списка |
| Обогащение справочников | Запоминание в голове или Excel | Автоматическое сохранение новых пар region\_code → esr\_code в PostgreSQL |
| Формирование отчётов | Ручная заливка цветом в Excel | Автоматическая визуализация: красный — нет в DMS, зелёный — есть |

2.5 Ожидаемые результаты внедрения

|  |  |  |
| --- | --- | --- |
| № | Результат | Показатель |
| 1 | Сокращение времени на создание одной ТТ | С 1–2 минут до 30 секунд |
| 2 | Уменьшение количества ошибок при вводе | 100% исключение ошибок |
| 3 | Повышение прозрачности процессов | 100% логирование всех операций |
| 4 | Ускорение сверки накладных | С 6 часов до 15 минут |
| 5 | Готовность к масштабированию | Поддержка автономной работы через чат-бот |

2.6 Обоснование необходимости разработки

В настоящее время все процессы выполняются вручную, что приводит к:

- Высокой трудоёмкости

- Ошибкам при вводе данных

- Задержкам в обработке запросов

- Отсутствию единого источника данных

- Сложности в обучении новых сотрудников

Автоматизация позволит:

- Освободить аналитика для более стратегических задач

- Обеспечить единообразие и качество данных

- Создать основу для дальнейшего развития

3. ТРЕБОВАНИЯ К ФУНКЦИОНАЛЬНЫМ ХАРАКТЕРИСТИКАМ

3.1 Общие положения

Система должна обеспечивать автоматизацию следующих ключевых функций:

1. Создание торговой точки (ТТ) в CRM на основе данных из 1С или текстового запроса.

2. Перепривязка ТТ к новому ESR по XCRM GUID с сохранением MRC-привязок.

3. Сверка накладных с системой DMS на основе отфильтрованного списка из Excel.

4. Поддержка AI-агента, способного извлекать данные из текста, уточнять недостающие поля и формировать JSON для запуска процессов.

5. Фоновая обработка процессов без блокировки графического интерфейса.

6. Логирование операций и ошибок в файл и базу данных.

7. Экспорт результатов в Excel с визуализацией статусов.

Система должна поддерживать два режима ввода:

- Ручной ввод через GUI (вкладки: "Создание ТТ", "Сверка накладных").

- AI-чат: текстовый запрос → извлечение данных → уточнение → запуск процесса.

3.2 Функциональные требования

Ниже представлены функциональные требования (FR), сформулированные в соответствии с ГОСТ, с трассировкой к user story и задачам из `product\_backlog.xlsx`.

ID Требование Источник (User Story / Task) Категория

----

FR-01 Система должна позволять вручную заполнять форму создания ТТ в GUI, включая поля: название, ИНН, адрес, район маршрута, ESR US1.1, T1.1 Ввод данных

FR-02 Система должна заполнять форму в CRM через Selenium, используя данные из GUI или AI-чата US1.2, T1.2 Автоматизация

FR-03 Система должна запускать процесс создания ТТ в фоновом режиме (без блокировки GUI) US1.3, T1.4 Производительность

FR-04 Система должна уведомлять пользователя о завершении процесса (успех/ошибка) через всплывающее окно US1.4, T1.5 Уведомления

FR-05 Система должна логировать все операции и ошибки в файл `logs/app.log` и в PostgreSQL US1.5, T1.6 Надёжность

FR-06 AI должен предсказывать атрибуты ТТ (вид, тип ассортимента, площадь, местоположение) на основе истории созданных точек US1.6, T1.12 AI-логика

FR-07 AI должен уточнять недостающие данные (например, ESR) до запуска процесса через диалог в чате US1.7, US3.2 AI-взаимодействие

FR-08 AI должен сохранять новые пары `region\_code → esr\_code` в PostgreSQL при ручном указании пользователем US1.8, T1.8 Хранение данных

FR-09 AI должен формировать JSON с данными о ТТ на основе текстового запроса в чате US1.9, T1.12 AI-логика

FR-10 Система должна создавать объекты `TradePoint` на основе JSON, сгенерированного AI US1.10, T1.1 Модель данных

FR-11 Система должна проверять наличие обязательных полей (`owner`, `address`, `region\_code`) перед запуском процесса US1.11, T1.10 Валидация

FR-12 Обязательные поля при создании ТТ: владелец (контрагент), адрес, район маршрута US1.12 Ввод данных

FR-13 Выходной результат создания ТТ должен включать отчётный Excel с полями: XCRM GUID, статус, ошибка (если есть) US1.13, T1.13 Экспорт

FR-14 Система должна обновлять сводные таблицы в Excel (аналог "Обновить данные") при загрузке файла US2.1, T1.9 Обработка Excel

FR-15 Система должна фильтровать накладные по условиям: `SFA = пусто`, `ISA ≠ 0` US2.2 Сверка

FR-16 Система должна копировать отфильтрованные накладные на отдельный лист для обработки US2.3 Обработка Excel

FR-17 Система должна заливать красным цветом накладные, которых нет в DMS US2.4, T1.13 Визуализация

FR-18 Система должна оставлять без изменений накладные, которые есть в DMS US2.5 Сверка

FR-19 Результат сверки должен сохраняться в Excel с пометками: "есть в DMS", "нет в DMS" US2.6, T1.13 Экспорт

FR-20 Процесс сверки должен выполняться в фоне без блокировки GUI US2.7, T1.4 Производительность

FR-21 Система должна перепривязывать ТТ к новому ESR по XCRM GUID US3.1, T1.2 Автоматизация

FR-22 При перепривязке система должна удалять только ESR-привязки, не затрагивая MRC-привязки US3.2, T1.2 Безопасность

FR-23 Система должна гарантированно перепривязывать территорию, даже если старый и новый ESR совпадают US3.3 Надёжность

FR-24 Данные о ТТ должны храниться в PostgreSQL для быстрого поиска и фильтрации US4.1, T1.7 Хранение данных

FR-25 Маппинг `район → ESR` должен храниться в PostgreSQL и синхронизироваться с Excel US4.3, T1.8 Хранение данных

FR-26 GUI должен содержать иконки на кнопках (загрузка, запуск, чат) без текста US5.1, T1.11 UX

FR-27 Система должна использовать дефолтные значения: `point\_type = "Counter"`, `assortment = "Весь ассортимент"`, `area = "1-50"`, `location\_type = "Street"`, `chain\_name = "NO CHAIN"` US5.2 Удобство

3.3 Бизнес-правила

№ Бизнес-правило Описание

BR-1 Формирование названия ТТ Поле `name` формируется как `{owner} {address}`

BR-2 Разделение процессов Наличие XCRM GUID → перепривязка; отсутствие → создание

BR-3 Однозначность ESR Каждый `region\_code` однозначно определяет `esr\_code` (если есть в БД)

BR-4 Направление деятельности После ввода ESR-кода система предлагает одно направление деятельности

BR-5 Защита MRC-привязок При перепривязке удаляются только строки с "ESR" в названии привязки

BR-6 Дефолтные значения Если поле не указано — используется дефолт (см. FR-27)

BR-7 Валидация перед запуском Процесс не запускается, если не хватает обязательных полей

BR-8 Обогащение справочника При ручном указании нового `region\_code → esr\_code` пара сохраняется в PostgreSQL

BR-9 Фоновая обработка Все процессы (создание, сверка) выполняются в `QThread`

BR-10 Цветовая визуализация Накладные, отсутствующие в DMS, заливаются красным цветом

3.4 Потоки данных

3.4.1 Поток: Создание новой ТТ

```mermaid

graph TD

A[Менеджер отправляет запрос] --> B{Есть XCRM GUID?}

B -- Нет --> C[AI извлекает: owner, address, region\_code]

C --> D[Поиск esr\_code по region\_code в PostgreSQL]

D --> E[Применение дефолтов: point\_type, area и др.]

E --> F[Формирование TradePoint]

F --> G[Валидация обязательных полей]

G --> H[Запуск процесса в фоне]

H --> I[Selenium: заполнение формы в CRM]

I --> J[Первичное сохранение → генерация XCRM GUID]

J --> K[Привязка к ESR: ввод кода → выбор направления]

K --> L[Финальное сохранение]

L --> M[Экспорт отчёта в Excel]

M --> N[Уведомление: успех]

```

3.4.2 Поток: Перепривязка ТТ

```mermaid

graph TD

A[Менеджер отправляет запрос с XCRM GUID] --> B[AI распознаёт: перепривязка]

B --> C[Selenium: поиск ТТ по XCRM GUID]

C --> D[Открытие формы редактирования]

D --> E[Анализ таблицы привязок]

E --> F{Строка содержит "ESR"?}

F -- Да --> G[Удалить (деактивировать)]

F -- Нет --> H[Сохранить]

G --> I[Добавление новой привязки: ESR + направление]

I --> J[Сохранение изменений]

J --> K[Экспорт отчёта]

K --> L[Уведомление: успех]

```

3.4.3 Поток: Сверка накладных

```mermaid

graph TD

A[Загрузка Excel-файла] --> B[Обновление сводных таблиц (COM/xlwings)]

B --> C[Фильтрация: SFA=пусто, ISA≠0]

C --> D[Копирование на отдельный лист]

D --> E[Запуск сверки в фоне]

E --> F{Накладная есть в DMS?}

F -- Да --> G[Оставить без изменений]

F -- Нет --> H[Залить красным цветом]

H --> I[Добавить пометку: "нет в DMS"]

G --> I

I --> J[Сохранение результата в Excel]

J --> K[Уведомление: завершено]

```

3.5 Структура сущности `TradePoint`

Поле Тип Обязательность Дефолт Источник

-----

`owner` str Да — 1С / запрос

`address` str Да — 1С / запрос

`region\_code` str Да — 1С / запрос

`inn` str Нет None 1С

`point\_type` str Нет "Counter" AI / дефолт

`assortment` str Нет "Весь ассортимент" дефолт

`location\_type` str Нет "Street" дефолт

`area` str Нет "1-50" дефолт

`chain\_name` str Нет "NO CHAIN" дефолт

`esr\_code` str Да (до привязки) — PostgreSQL / ручной ввод

`xcrm\_guid` str После создания — CRM

`status` str — "pending" Система

`error\_message` str — None Система

> Вычисляемое поле: `name = owner + " " + address`

3.6 Триггеры и реакции системы

Триггер Реакция системы

--

Запрос в чате: "Создать ТТ: ООО Ромашка, Новосибирск, ул. Ленина, 15, район Н02" AI извлекает данные, проверяет наличие `region\_code → esr\_code`, запрашивает уточнения (если нужно), формирует JSON

Пользователь нажимает кнопку "Запустить" на вкладке "Создание ТТ" Проверка обязательных полей → запуск процесса в фоне → Selenium заполняет CRM

Загружен Excel с накладными Обновление данных → фильтрация → копирование → запуск сверки

Сервер DMS недоступен Ошибка логируется, процесс продолжается для следующей накладной

Процесс завершён (успех/ошибка) Показ уведомления, экспорт отчёта, обновление статуса

4. ТРЕБОВАНИЯ К НАДЁЖНОСТИ

4.1 Общие положения

Система должна обеспечивать высокую степень надёжности при выполнении критически важных операций: создание торговых точек, перепривязка ESR, сверка накладных.

Она должна быть устойчива к:

- Временным сбоям внешних систем (CRM, DMS),

- Ошибкам пользователя,

- Потере данных,

- Прерыванию процессов.

Система должна не допускать потери данных, не нарушать целостность привязок (особенно MRC), и предоставлять полную информацию о причинах сбоев.

4.2 Отказоустойчивость

ID Требование Источник Категория

---

NFR-REL-01 Система должна продолжать работу при временной недоступности DMS (например, сервер упал, таймаут) T1.14, US5.4 Обработка ошибок

NFR-REL-02 При недоступности DMS система должна логировать ошибку и переходить к следующей накладной, не прерывая процесс сверки T1.14, US5.4 Обработка ошибок

NFR-REL-03 Система должна обрабатывать капчу, редиректы и смену сессии в CRM/DMS, повторяя попытку входа при необходимости US4.4, T1.2 Стабильность Selenium

NFR-REL-04 При падении процесса в фоне (QThread) система должна корректно завершить поток и зафиксировать ошибку в логах T1.4 Безопасность

NFR-REL-05 Система должна поддерживать повторные попытки (retry) при сетевых ошибках (например, 502, 503) с экспоненциальной задержкой T1.14 Надёжность

4.3 Сохранность данных

ID Требование Источник Категория

---

NFR-REL-06 Все операции и ошибки должны логироваться в файл `logs/app.log` с уровнем DEBUG US1.5, T1.6 Логирование

NFR-REL-07 Логи должны содержать: дату и время, тип операции, входные данные, статус (успех/ошибка), сообщение об ошибке, XCRM GUID (если есть) US1.5 Логирование

NFR-REL-08 Данные о ТТ, маппинге `region\_code → esr\_code` и результатах сверки должны храниться в PostgreSQL с регулярным резервным копированием US4.1, T1.7 Хранение данных

NFR-REL-09 Перед запуском процесса система должна проверять наличие обязательных полей и не начинать операцию при их отсутствии US1.11, T1.10 Защита от ошибок

NFR-REL-10 При перепривязке ESR система не должна удалять привязки к MRC, даже если они находятся в одной таблице US3.2, T1.2 Целостность данных

4.4 Восстановление после сбоев

ID Требование Источник Категория

---

NFR-REL-11 После сбоя (например, падение DMS) система должна продолжать обработку следующих элементов списка (накладных, ТТ) T1.14 Продолжение работы

NFR-REL-12 При сбое в процессе создания ТТ система должна сохранить частичные данные (например, заполненные поля) для последующего дозапуска US1.13 Восстановление

NFR-REL-13 При аварийном завершении приложения система должна сохранять состояние последних операций в БД для аудита US4.1 Аудит

NFR-REL-14 При перезапуске приложения система должна синхронизировать маппинг `region\_code → esr\_code` из Excel в PostgreSQL US4.3, T1.8 Синхронизация

4.5 Обработка ошибок

ID Требование Источник Категория

---

NFR-REL-15 Система должна валидировать JSON перед запуском процесса (обязательные поля, формат данных) US1.11, T1.10 Валидация

NFR-REL-16 При ошибке валидации процесс не должен запускаться, пользователь получает уведомление с описанием ошибки US1.11 Защита от сбоев

NFR-REL-17 При ошибке в CRM (например, дубль по адресу) система должна зафиксировать ошибку в логе и отчёте, не прерывая общего процесса US1.5 Обработка исключений

NFR-REL-18 При ошибке в Excel (недоступен файл, пароль) система должна уведомить пользователя и предложить повторить попытку US2.1, T1.9 Обработка исключений

NFR-REL-19 При вводе некорректного ESR-кода система должна проверить его наличие в PostgreSQL и Excel, при отсутствии — запросить уточнение у пользователя US4.3, US3.7 Защита от ошибок

4.6 Бизнес-правила надёжности

№ Правило Описание

--

BR-REL-1 Нет прерывания из-за одной ошибки При массовой обработке (ТТ, накладные) ошибка одной записи не должна останавливать весь процесс

BR-REL-2 Приоритет целостности MRC Удаление привязок разрешено только для тех, чьё название содержит "ESR" или "territory"

BR-REL-3 Обязательная валидация до запуска Процесс не запускается, если не хватает `owner`, `address`, `region\_code`

BR-REL-4 Фоновая обработка без блокировок GUI не должен зависать при ошибках в Selenium или DMS

BR-REL-5 Повторная привязка = успех Даже если старый и новый ESR совпадают — система должна выполнить полный цикл перепривязки

4.7 Схема обработки сбоев

```mermaid

graph TD

A[Запуск процесса] --> B{Валидация данных}

B -- Ошибки --> C[Показать уведомление, не запускать]

B -- OK --> D[Запуск в фоне (QThread)]

D --> E{Ошибка в CRM/DMS?}

E -- Да --> F[Логировать ошибку]

F --> G{Продолжить?}

G -- Да --> H[Перейти к следующей записи]

G -- Нет --> I[Завершить процесс]

E -- Нет --> J[Обработка успешна]

J --> K[Обновить статус, GUID]

K --> L[Экспорт отчёта]

L --> M[Показать уведомление]

```

> Ключевые принципы:

> - Ошибки не прерывают массовую обработку

> - Все сбои логируются и видны пользователю

> - Целостность данных — приоритет №1

- Пункт 4.8 "Требования к резервному копированию" исключён из текущей версии документа.

- Пункт 4.9 "Требования к восстановлению" также исключён, так как он напрямую связан с резервным копированием.

- Удаление этих требований обусловлено ограниченными рамками MVP и планами по интеграции с 1С, которая в будущем станет единым источником данных.

- Временное хранение данных (PostgreSQL, Excel) рассматривается как прототипное, не требующее жёстких мер по резервированию на этапе первого спринта.

5. УСЛОВИЯ ЭКСПЛУАТАЦИИ

5.1 Общие положения

Система предназначена для эксплуатации на персональных компьютерах аналитиков отдела продаж в рамках локальной сети компании.

Работа системы осуществляется в автономном режиме с периодическим подключением к внешним системам:

- CRM (через веб-интерфейс)

- DMS (через веб-интерфейс)

- Excel-файлы (локальные или сетевые)

Система не требует постоянного подключения к серверу приложений и не является веб-сервисом.

5.2 Требования к аппаратному обеспечению

Параметр Минимальные требования Рекомендуемые требования

----

Процессор Intel Core i3 или аналог Intel Core i5 / i7 или выше

Оперативная память 8 ГБ 16 ГБ

Свободное место на диске 5 ГБ 10 ГБ

Видеокарта Интегрированная Интегрированная

Монитор Разрешение 1366×768 Разрешение 1920×1080

Система не является ресурсоёмкой, но требует достаточной памяти для обработки Excel-файлов и одновременной работы с браузерами (Selenium).

5.3 Требования к программному обеспечению

Компонент Требование

---

Операционная система Windows 10, Windows 11 (64-bit)

Интерпретатор Python Python 3.10 – 3.12

Библиотеки Python PyQt5, selenium, pandas, openpyxl, xlwings, psycopg2, pydantic

Браузер Google Chrome (актуальная версия)

Microsoft Excel Excel 2016 или новее (для COM-доступа)

PostgreSQL PostgreSQL 14+ (локально или в сети)

Дополнительно ChromeDriver (совместимый с версией Chrome)

5.4 Сетевые требования

Подключение к интернету - Обязательно (для доступа к CRM и DMS)

Скорость соединения - Минимум 10 Мбит/с (рекомендуется 50 Мбит/с)

Доступ к CRM - Через корпоративную сеть

Доступ к DMS - Через корпоративную сеть

Доступ к PostgreSQL - Локально или через внутреннюю сеть

Доступ к файлам - Локальные или сетевые пути

5.5 Ограничения по использованию

Ограничение Описание

--

Многопользовательский режим Не поддерживается. Система рассчитана на одного пользователя на рабочей станции

Кроссплатформенность Не поддерживается. Только Windows (из-за Excel + COM)

Автономная работа Возможна, но с ограничениями: без доступа к CRM/DMS/Excel — только просмотр логов и истории

Масштабирование На текущем этапе — нет. В будущем — через интеграцию с 1С и чат-бот

Резервное копирование Не реализовано. Данные в PostgreSQL и Excel считаются временными до интеграции с 1С

Что мы удаляем из раздела 5:

Элемент Причина удаления

--

Пункт 5.4 "Сетевые требования" — упоминание VPN Вы пояснили: доступ к CRM и DMS — через внутреннюю сеть, VPN не требуется и может мешать.

Пункт 5.6 "Условия хранения и эксплуатации" Это требования к физической среде (температура, влажность и т.д.) — не относятся к ПО, это избыточно для ТЗ на программный продукт.

Пункт 5.7 "Гарантийные обязательства" Применимо к коммерческим продуктам, но не к внутреннему ПО, разрабатываемому аналитиком для своих задач.

6. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

6.1 Общие положения

Система должна сопровождаться полным комплектом программной документации, соответствующей ГОСТ 19.503-79 и ГОСТ 19.504-79, адаптированным под внутреннюю разработку.

Документация предназначена для:

- Аналитика (пользователь, разработчик, поддержка),

- IT-специалистов (администрирование, мониторинг),

- Будущих разработчиков (расширение функциональности).

Документация должна быть доступна в электронном виде и поддерживаться в актуальном состоянии на всех этапах разработки.

6.2 Перечень программной документации

№ Наименование документа Назначение Формат Ответственный

-----

1 Техническое задание (ТЗ) Описание требований к системе `.docx`, `.pdf` Бизнес-аналитик

2 Руководство пользователя Пошаговые инструкции для аналитика `.pdf`, `.html` Бизнес-аналитик

3 Описание программы Структура модулей, архитектура, POM `.md`, `.pdf` Разработчик

4 Описание применения Сценарии использования, потоки данных `.pdf`, `.drawio` Бизнес-аналитик

5 Программа и методика испытаний (ПМИ) Тест-кейсы, сценарии проверки `.xlsx`, `.pdf` QA / Аналитик

6 Отчёт о приёмо-сдаточных испытаниях Результаты тестирования `.pdf` Аналитик

6.3 Требования к содержанию документации

6.3.1 Руководство пользователя

Должно включать:

- Как запустить приложение

- Описание вкладок GUI: "Создание ТТ", "Сверка накладных", "AI-чат"

- Как вручную заполнить форму создания ТТ

- Как загрузить Excel с накладными

- Как взаимодействовать с AI-агентом (примеры запросов)

- Как интерпретировать результаты (отчёты, уведомления, логи)

- Что делать при ошибках

6.3.2 Описание программы

Должно включать:

- Архитектуру: operation-process-view

- Структуру проекта:

```

project/

├── application/ Инициализация GUI

├── domain/ Сущности (TradePoint)

├── infrastructure/ Selenium, Excel, DB

├── operation/ Модули: create\_tt, reconcile\_invoices

├── process/ Обработка данных

├── use\_cases/ Сценарии использования

├── ai\_agent/ Логика LLM

└── config/ Конфиги

```

- Диаграммы:

- POM (Page Object Model) для CRM и DMS

- Потоки данных (создание ТТ, сверка)

- Диаграмма классов (TradePoint)

6.3.3 Описание применения

Должно включать:

- Сценарии использования:

1. Создание ТТ по запросу менеджера

2. Перепривязка ТТ к новому ESR

3. Сверка накладных из Excel

4. Взаимодействие с AI-агентом

- Потоки данных (см. раздел 3)

- Правила обработки:

- Как определяется, нужно ли создавать или перепривязывать

- Как формируется `name` ТТ

- Как обрабатываются ошибки

6.3.5 Программа и методика испытаний

№ Сценарий Шаги Ожидаемый результат

1 Создание ТТ через GUI Заполнить форму → нажать "Запустить" ТТ создана в CRM, GUID в отчёте

2 Перепривязка ТТ Ввести XCRM GUID → указать новый ESR → запустить ESR-привязка обновлена, MRC не удалена

3 Сверка накладных Загрузить Excel → запустить сверку Накладные без DMS помечены красным

4 AI-запрос: создание ТТ Написать: "Создать ТТ: ООО Ромашка, Новосибирск, ул. Ленина, 15, район Н02" AI формирует JSON → запускает процесс

5 Отсутствует ESR AI не находит `region\_code` в БД AI уточняет: "Не найден ESR для Н02. Указать?"

6 Ошибка в DMS Сервер DMS недоступен Ошибка логируется, процесс продолжается

7 Валидация JSON Передать JSON без `owner` Процесс не запускается, ошибка в логе

6.4 Форматы и средства оформления

Документ Формат Средства

----

ТЗ, Руководства `.docx`, `.pdf` Microsoft Word / LibreOffice

Описание программы `.md`, `.pdf` Markdown, draw.io, PlantUML

Диаграммы `.svg`, `.png` draw.io, Mermaid, Lucidchart

ПМИ `.xlsx`, `.pdf` Excel, Google Sheets

6.5 Требования к актуализации документации

Событие Действие

--

Добавление новой функции Обновить ТЗ, Руководство пользователя, Описание программы

Изменение структуры БД Обновить Описание программы и Руководство администратора

Смена URL CRM/DMS Выпустить быстрый hotfix коммит приложения и обновить версию всем пользователям

Завершение спринта Провести ревизию всей документации

7. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

7.1 Общие положения

Система предназначена для работы в локальной среде аналитика и не требует серверной инфраструктуры.

Техническое обеспечение включает:

- Аппаратные компоненты (уже описаны в разделе 5)

- Программные компоненты

- Внешние системы и точки интеграции

- Инструменты разработки и сопровождения

Система предназначена для локального использования аналитиком отдела продаж.

Все конфигурационные данные хранятся локально в приложении, доступны через вкладку "Настройки" в GUI.

Никакие конфиденциальные данные (логины, пароли, API-ключи) не сохраняются в репозитории или внешних файлах.

Система не требует серверной инфраструктуры и работает в автономном режиме с доступом к внешним веб-системам.

7.2 Программные компоненты системы

Все зависимости будут указаны в файле `requirements.txt`, который не включается в документацию, но является частью релиза.

Компонент Назначение Версия / Требование

---

Python Основной язык программирования 3.10 – 3.12

PyQt5 Графический интерфейс (GUI) >= 5.15

Selenium Автоматизация браузера (CRM, DMS) >= 4.0

pandas Обработка данных Excel >= 1.3

openpyxl Чтение/запись Excel (без COM) >= 3.0

xlwings Работа с Excel через COM (обновление данных) >= 0.27

psycopg2 Подключение к PostgreSQL >= 2.9

pydantic Валидация данных, JSON >= 1.10

python-dotenv Загрузка переменных окружения (опционально) >= 0.21

7.3 Внешние системы и точки интеграции

> - OpenRouter API используется как единая точка доступа к LLM (Qwen, GPT и др.)

> - Пользователь может выбрать модель, изменить API-ключ, обновить URL API

> - При первом запуске — обязательное заполнение настроек для активации функциональности

Система Тип доступа Протокол Назначение

-----

CRM Веб-интерфейс HTTPS Создание ТТ, привязка ESR

DMS Веб-интерфейс HTTPS Проверка наличия накладных

1С Файловый обмен (будущее) SMB / API Получение данных о ТТ

Microsoft Excel Локальный файл COM / openpyxl Загрузка и обработка данных

PostgreSQL Локальная / сетевая БД TCP/IP Хранение данных о ТТ и маппинге

OpenRouter API API HTTPS Парсинг запросов от пользователя, генерация JSON для запуска процессов

7.4 Модель хранения настроек

Настройки системы хранятся локально на устройстве пользователя в виде сериализованной структуры (например, `settings.json` или pickle), не выгружается в репозиторий.

Поля настроек:

Группа Параметр Описание

-

Авторизация CRM: логин, пароль Вводится пользователем

DMS: логин, пароль Вводится пользователем

API Интеграции OpenRouter: API URL По умолчанию: `https://openrouter.ai/api/v1`

OpenRouter: API Key Вводится пользователем

OpenRouter: Модель Выбор из списка (Qwen, GPT, Claude и др.)

Системные ссылки CRM URL Жёстко задан в коде, может быть обновлён в hotfix

DMS URL Жёстко задан в коде, может быть обновлён в hotfix

Файловые пути Входной Excel Фиксированный путь: `C:\Data\input.xlsx`

Выходной отчёт Фиксированный путь: `C:\Data\output.xlsx`

1С сервер Адрес сервера 1С (будущее)

> Поведение при первом запуске:

> - Пользователь перенаправляется на вкладку "Настройки"

> - Не может использовать функции, зависящие от конфигурации, пока не заполнит:

> - CRM/DMS логин/пароль

> - OpenRouter API Key

> - В интерфейсе есть кнопка "Получить API Key" → открывает браузер с `https://openrouter.ai/keys`

Система Механизм обновления

--

CRM URL Жёстко задан в коде → обновляется в `hotfix`-ветке при изменении

DMS URL То же

OpenRouter API URL Может быть изменён пользователем в настройках

> Процедура:

> При поступлении жалобы на недоступность системы → разработчик вносит правку в URL → собирает новую версию → выпускает в тот же день.

7.5 Пути к файлам

Тип файла Путь Примечание

---

Входной Excel `C:\Data\input.xlsx` Автоматически загружается при старте процесса

Выходной отчёт `C:\Data\output.xlsx` Открывается автоматически после завершения процесса

Логи `logs/app.log` Создаётся при первом запуске

> Автоматическое открытие:

> После завершения любого процесса (создание ТТ, сверка) — выходной файл открывается автоматически через `os.startfile()`.

7.6 Инструменты разработки и сопровождения

Инструмент Назначение

----

PyCharm / VS Code IDE для разработки

Git Управление версиями (репозиторий)

Jira / Notion Управление задачами и бэклогом

draw.io / Lucidchart Диаграммы (POM, потоки данных)

Postman Тестирование API (LLM, 1С в будущем)

DBeaver / pgAdmin Работа с PostgreSQL

Confluence / Notion Документация

7.7 Управление настройками через GUI

Элемент Описание

--

Вкладка "Настройки" Централизованное управление всеми параметрами

Поле ввода API Key С маской скрытия пароля

Кнопка "Проверить подключение" Для CRM, DMS, OpenRouter

Кнопка "Открыть OpenRouter" Переход на сайт для получения ключа

Сохранение Локально, без шифрования (в защищённой среде пользователя)

7.8 Требования к внешним системам

Система Требование

----

CRM Доступ по логину/паролю, поддержка Selenium (не блокирует автоматизацию)

DMS Стабильная работа, поддержка поиска по номеру накладной

PostgreSQL Доступ на чтение/запись для пользователя `assistant`

Excel Файлы без макросов, с разрешённым доступом через COM (если используется `xlwings`)

Браузер Google Chrome (актуальная версия), установленный `chromedriver`

7.9 Архитектура интеграции

```mermaid

graph LR

A[Пользователь] --> B[GUI (PyQt5)]

B --> C[AI-Agent (LLM)]

C --> D[Data Processing]

D --> E[Operation: create\_tt / reconcile]

E --> F[Selenium → CRM]

E --> G[Selenium → DMS]

D --> H[PostgreSQL]

B --> I[Excel (openpyxl / xlwings)]

H --> J[Кэш для AI]

I --> J

J --> C

```

> Ключевые принципы:

> - Все внешние вызовы — через модули `infrastructure`

> - Данные — через PostgreSQL как единый источник правды

> - GUI — не зависит напрямую от Selenium или Excel

8. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

8.1 Общие положения

Система использует структурированные и полуструктурированные данные, поступающие из внешних и внутренних источников.

Основные цели информационного обеспечения:

- Обеспечение единого источника правды для справочников (например, `region\_code → esr\_code`)

- Поддержка быстрого поиска и фильтрации данных о торговых точках

- Хранение истории операций для анализа и отладки

- Временное хранение входных и выходных файлов в формате Excel

Информационное обеспечение включает:

- Базу данных PostgreSQL

- Локальные Excel-файлы

- Внутренние структуры данных (например, `TradePoint`)

- Конфигурационные данные (хранятся в GUI)

8.2 Источники данных

Источник Тип Формат Назначение Частота обновления

---------

1С (будущее) Внешняя система API / SQL Получение данных о ТТ, контрагентах, маршрутах Ежедневно

CRM Веб-интерфейс HTML Создание ТТ, привязка ESR В реальном времени

DMS Веб-интерфейс HTML Проверка наличия накладных В реальном времени

Excel (входной) Локальный файл `.xlsx` Список накладных для сверки При каждом запуске

Excel (справочник) Локальный файл `.xlsx` Маппинг `region\_code → esr\_code` При старте системы

PostgreSQL Локальная / сетевая БД Реляционная Хранение данных о ТТ, маппинге, логах Постоянно

OpenRouter API Внешний API JSON Парсинг запросов, генерация JSON При каждом запросе в чате

8.3 Структура базы данных

База данных реализована в PostgreSQL 14+ и содержит следующие таблицы:

8.3.1 Таблица `trade\_points`

Поле Тип Описание Обязательность

---

`id` SERIAL Уникальный идентификатор Да

`owner` VARCHAR(255) Контрагент (владелец ТТ) Да

`address` TEXT Адрес торговой точки Да

`region\_code` VARCHAR(10) Код района маршрута (например, Н02) Да

`inn` VARCHAR(12) ИНН контрагента Нет

`point\_type` VARCHAR(50) Вид ТТ (по умолчанию: "Counter") Нет

`assortment` VARCHAR(100) Тип ассортимента (по умолчанию: "Весь ассортимент") Нет

`location\_type` VARCHAR(50) Местоположение (по умолчанию: "Street") Нет

`area` VARCHAR(20) Площадь (по умолчанию: "1-50") Нет

`chain\_name` VARCHAR(100) Название сети (по умолчанию: "NO CHAIN") Нет

`esr\_code` VARCHAR(20) Код ESR для привязки Да (до привязки)

`xcrm\_guid` UUID Уникальный идентификатор ТТ в CRM После создания

`status` VARCHAR(20) Статус: pending, created, failed, reattached Да

`error\_message` TEXT Описание ошибки (если есть) Нет

`created\_at` TIMESTAMP Дата и время создания Да

`updated\_at` TIMESTAMP Дата и время последнего изменения Да

> Индексы:

> - `idx\_xcrm\_guid` — по `xcrm\_guid`

> - `idx\_region\_code` — по `region\_code`

> - `idx\_status` — по `status`

8.3.2 Таблица `region\_esr\_mapping`

Поле Тип Описание Обязательность

---

`id` SERIAL Уникальный идентификатор Да

`region\_code` VARCHAR(10) Код района (например, Н02) Да

`esr\_code` VARCHAR(20) Соответствующий ESR-код Да

`direction` VARCHAR(100) Направление деятельности (например, "ООО Континент 'Абакан'") Да

`created\_at` TIMESTAMP Дата добавления Да

`updated\_at` TIMESTAMP Дата обновления Да

> Уникальность:

> - Ограничение `UNIQUE(region\_code)` — один район → один ESR

> - Синхронизация с Excel при старте системы (см. US4.3, T1.8)

8.3.3 Таблица `invoice\_reconciliation`

Поле Тип Описание Обязательность

---

`id` SERIAL Уникальный идентификатор Да

`invoice\_number` VARCHAR(50) Номер накладной Да

`isa\_value` DECIMAL Значение ISA Да

`sfa\_value` TEXT Значение SFA (пусто / число) Да

`in\_dms` BOOLEAN Есть ли накладная в DMS Да

`check\_date` TIMESTAMP Дата проверки Да

`error\_message` TEXT Описание ошибки (если DMS недоступен) Нет

> Индексы:

> - `idx\_invoice\_number` — для быстрого поиска

> - `idx\_check\_date` — для анализа по периодам

8.4 Форматы входных и выходных данных

8.4.1 Входные данные

Источник Формат Структура

-----

Excel (маппинг) `.xlsx` Таблица: `Район маршрута`, `ESR-код`, `Направление деятельности` — используется для синхронизации с `region\_esr\_mapping` в PostgreSQL

Excel (контрагенты) `.xlsx` Таблица: `ИНН`, `Название`, `Текущая сеть` — используется для заполнения `contractors`

Маппинг — это таблица соответствий:

Район маршрута ESR-код Направление деятельности

-----

Н02 ESR-123 ООО Континент "Абакан"

Н03 ESR-456 ООО Континент "Новосибирск"

Эта таблица хранится в Excel-файле, например: `mapping\_region\_to\_esr.xlsx`.

Таблица: `contractors` (контрагенты и их сети)

Поле Тип Описание

--

`id` SERIAL Уникальный идентификатор

`inn` VARCHAR(12) ИНН контрагента

`name` VARCHAR(255) Название контрагента

`current\_chain` VARCHAR(100) Текущее название сети (например, "Wholesale")

`last\_updated` TIMESTAMP Дата последнего обновления

`is\_active` BOOLEAN Активен ли контрагент

> Источник данных:

> - Первоначально — из Excel-файла `contractors.xlsx`

> - В будущем — из 1С

🔄 Логика обработки при создании/обновлении ТТ

```mermaid

graph TD

A[Получен запрос на создание ТТ] --> B{Есть XCRM GUID?}

B -- Да --> C[Проверить: нужно ли перепривязать?]

C --> D[AI уточняет: "Нужно изменить сеть?" если изменился current\_chain]

D --> E{Нужно изменить сеть?}

E -- Да --> F[Деактивировать старую ТТ в CRM]

F --> G[Создать новую ТТ с новым chain\_name]

E -- Нет --> H[Перепривязать ESR]

B -- Нет --> I[Проверить контрагента в БД]

I --> J{Есть в contractors?}

J -- Да --> K[Использовать current\_chain]

J -- Нет --> L[Запросить chain\_name у пользователя]

K --> M[Модель предсказывает атрибуты]

M --> N[Создать ТТ]

```

Почему это важно?

1. Предотвращает ошибки в отчётах партнёров

Если сеть указана неверно — отчёт партнёра будет некорректным.

2. Поддерживает бизнес-логику

CRM не позволяет менять `chain\_name` — значит, нужно пересоздавать ТТ, а не редактировать.

3. Готовит систему к интеграции с 1С

В 1С будет актуальная информация о смене формата — и система сможет реагировать автоматически.

🔄 Как система использует этот файл?

1. При запуске приложения:

- Система читает `mapping\_region\_to\_esr.xlsx`

- Загружает данные в таблицу `region\_esr\_mapping` в PostgreSQL

2. При создании ТТ:

- Пользователь указывает `region\_code = Н02`

- Система ищет `esr\_code` в `region\_esr\_mapping`

- Если нашла — подставляет

- Если не нашла — AI уточняет у пользователя

3. При ручном указании ESR:

- Пара `region\_code → esr\_code` сохраняется в PostgreSQL

- При следующем запуске — синхронизируется обратно в Excel (если нужно)

> Сделать Excel источником истины на старте, но PostgreSQL — операционной базой для поиска и обогащения.

8.4.2 Выходные данные

Назначение Формат Структура

----

Отчёт о создании ТТ `.xlsx` Колонки: `XCRM GUID`, `Статус`, `Ошибка (если есть)`

Отчёт о сверке накладных `.xlsx` Исходный файл с цветовой заливкой: красный — нет в DMS

Логи `.log` Строки: `timestamp level operation status message`

Внутренний JSON (для процессов) `dict` `{ "owner": "...", "address": "...", "region\_code": "Н02", ... }`

> Автоматическое открытие:

> После завершения процесса выходной Excel-файл открывается автоматически через `os.startfile()`.

8.5 Правила обработки данных

Правило Описание

--

BR-DATA-1 При создании ТТ поле `name` формируется как `{owner} {address}`

BR-DATA-2 При отсутствии `esr\_code` система ищет его в `region\_esr\_mapping` по `region\_code`

BR-DATA-3 Если `region\_code` не найден в БД, AI уточняет у пользователя

BR-DATA-4 При ручном указании `esr\_code` пара `region\_code → esr\_code` сохраняется в БД

BR-DATA-5 При перепривязке удаляются только привязки, содержащие "ESR" в названии

BR-DATA-6 Дефолтные значения применяются, если поля не указаны: `point\_type = "Counter"` и др.

BR-DATA-7 Процесс не запускается при отсутствии: `owner`, `address`, `region\_code`

BR-DATA-8 Все операции логируются в `logs/app.log` и в `trade\_points.error\_message`

BR-DATA-9 Если `point\_type`, `assortment`, `location\_type`, `area` не указаны — система должна использовать модель классификации для предсказания на основе истории и контекста

BR-DATA-10 Дефолтные значения (`Counter`, `1-50`, `Street`) используются только если модель не может сделать предсказание

BR-DATA-11 При создании ТТ система проверяет `current\_chain` контрагента в таблице `contractors`

BR-DATA-12 Если `current\_chain` изменился по сравнению с существующей ТТ в CRM — система должна деактивировать старую ТТ и создать новую с новым `chain\_name`

Новое правило (корректное):

> "Если поля не указаны — система должна использовать модель классификации для предсказания атрибутов на основе:

> - истории созданных ТТ по этому контрагенту

> - географического положения

> - названия/адреса

> - типа сети (если известен)"

> Дефолтные значения используются только как резервный вариант, если модель не может сделать предсказание."

🧠 Архитектура модели классификации

Компонент Описание

-

Входные признаки (features) `owner`, `address`, `region\_code`, `inn`, `chain\_name` (если есть), исторические данные

Целевые переменные (target) `point\_type`, `assortment`, `location\_type`, `area`

Модель `scikit-learn` (Logistic Regression, Random Forest) или `XGBoost`

Хранение модели `.pkl` файл в папке `models/`

Обучение При добавлении новой ТТ — модель переобучается (или инкрементально обновляется)

Использование При создании новой ТТ — модель предсказывает атрибуты, если они не указаны

> Система обучается на ваших действиях, становится "умнее" с каждым днём.

8.6 Схема синхронизации данных

```mermaid

graph TD

A[Excel: region\_esr\_mapping.xlsx] -->При старте B(PostgreSQL: region\_esr\_mapping)

B --> C[AI-Agent: поиск ESR по region\_code]

C --> D[Process: создание ТТ / сверка]

D --> E[CRM / DMS]

D --> F[PostgreSQL: trade\_points, invoice\_reconciliation]

F --> G[Excel: отчёт (выходной)]

G --> H[Автоматическое открытие]

```

> Ключевые принципы:

> - PostgreSQL — единый источник правды для справочников

> - Excel — временный источник и приёмник

> - Синхронизация маппинга — при запуске приложения

8.7 Требования к сохранности данных

Угроза Мера защиты

--

Потеря `config.yaml` Не используется — настройки в GUI

Потеря `trade\_points` Восстановление из истории (до интеграции с 1С)

Потеря `region\_esr\_mapping` Восстановление из Excel-файла

Потеря входного Excel Повторная загрузка из исходного источника

Ошибка в DMS Логирование, продолжение обработки

> Примечание:

> На текущем этапе (MVP) система рассчитана на временное хранение данных.

> В будущем (после интеграции с 1С) 1С станет единым источником истины.

9. МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

9.1 Общие положения

Система не является публичным веб-приложением и не требует высокого уровня защиты по стандартам ПДн или ФСТЭК.

Однако, ввиду обработки конфиденциальных данных, необходимо обеспечить базовую защиту от несанкционированного доступа, утечки и модификации данных.

Цель раздела — определить минимально достаточные меры безопасности, соответствующие уровню риска и контексту использования.

9.2 Угрозы безопасности

Угроза Описание Уровень риска

----

Несанкционированный доступ к учетным данным Доступ к логинам и паролям от CRM/DMS при использовании чужой рабочей станции Высокий

Утечка API-ключей Получение доступа к OpenRouter API Key → использование за счёт компании Средний

Утечка данных о контрагентах Доступ к ИНН, адресам, названиям ТТ Средний

Модификация конфигурации Злоумышленник изменяет URL CRM/DMS → подмена данных Средний

Кража или копирование локальных файлов Доступ к `logs/app.log`, `output.xlsx`, `settings.json` Средний

Использование системы посторонним лицом Работа с приложением без аутентификации Высокий

9.3 Защита конфиденциальных данных

Данные Мера защиты Комментарий

-----

Логины и пароли от CRM/DMS Хранятся локально в зашифрованном виде (опционально) или в скрытом поле ввода в GUI Не сохраняются в файлах, не логируются

OpenRouter API Key Вводится пользователем, не логируется, хранится локально Рекомендуется использовать ключи с ограниченными правами

ИНН, адреса, названия ТТ Не являются персональными данными, но относятся к коммерческой тайне Доступ только авторизованным пользователям

`settings.json` (локальный файл) Хранится в папке пользователя, не экспортируется При утере — пользователь вводит данные заново

`logs/app.log` Не содержит паролей и API-ключей Логируются только операции, статусы, GUID

> Запрещено:

> - Логирование паролей, API-ключей, ИНН

> - Сохранение конфиденциальных данных в репозитории

> - Передача данных через незашифрованные каналы (кроме внутренней сети)

9.4 Средства защиты

Средство Описание Реализация

----

Аутентификация на уровне пользователя Система рассчитана на одного пользователя, но требует ввода учетных данных при первом запуске Настройки недоступны без заполнения обязательных полей

Шифрование (опционально) Возможность шифрования файла `settings.json` с помощью `Fernet` (cryptography) Реализуется в будущем при расширении на несколько пользователей

Ограничение доступа к файлам Пути к файлам (`C:\Data\...`) защищены правами ОС Только пользователь и администратор могут читать/писать

HTTPS для внешних API Все запросы к OpenRouter API выполняются по зашифрованному каналу Проверка SSL-сертификата

Очистка памяти Пароли и ключи после использования не хранятся в памяти дольше необходимого Использование `del` или контекстных менеджеров

Защита от XSS и инъекций Ввод через GUI контролируется, чат с AI не исполняет код Нет интерпретации пользовательского ввода как команд

9.5 Ограничения системы безопасности

Ограничение Обоснование

-

Отсутствие встроенной аутентификации (логин/пароль для приложения) Система рассчитана на одного пользователя — аналитика. Доступ к ПК контролируется на уровне ОС

Отсутствие шифрования `settings.json` На текущем этапе (MVP) — избыточно. Будет добавлено при переходе на многопользовательский режим

Отсутствие аудита действий пользователей Не требуется для внутреннего ПО. Логи операций хранятся в `app.log`

Отсутствие резервного копирования конфиденциальных данных Все данные могут быть восстановлены пользователем вручную (ввод логинов, ключей)

9.6 Рекомендации по безопасной эксплуатации

Рекомендация Описание

-

🔐 Не оставлять ПК без присмотра При использовании системы закрывать приложение или блокировать экран (Win+L)

🔄 Регулярно обновлять API-ключи Особенно OpenRouter — использовать ключи с ограниченным сроком действия

📁 Хранить файлы в защищённой папке `C:\Data\` — права доступа только для пользователя

🧹 Не делиться скриншотами с логами Даже если нет паролей — GUID и названия ТТ могут быть чувствительны

🛡️ Использовать антивирус и брандмауэр Для защиты от несанкционированного доступа к файлам

🌐 Не использовать VPN при работе с CRM/DMS Как указано: это может нарушить доступ к внутренним ресурсам

9.7 Обработка инцидентов

Сценарий Действие

--

Утеря доступа к ПК Заблокировать систему, изменить пароли от CRM/DMS при необходимости

Подозрение на утечку API-ключа Немедленно отозвать ключ на `openrouter.ai`, создать новый

Попытка несанкционированного доступа Зафиксировать в логах, сообщить в ИТ-службу

Кража ноутбука с установленной системой Удалить данные, изменить учетные данные, сообщить о потере

9.8 Требования к защите при хранении и передаче

Процесс Требование

----

Хранение паролей Не в открытом виде. В GUI — маскирование (•••••)

Хранение API-ключей Локально, без логирования

Передача данных в OpenRouter Только по HTTPS

Передача данных в CRM/DMS Через HTTPS, без кэширования паролей в браузере

Экспорт отчётов Без паролей и ключей. Только GUID, статусы, номера накладных

10. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

10.1 Общие положения

Разработка системы осуществляется по итеративной модели с использованием подхода Scrum.

Каждая итерация (спринт) длится 2 недели и завершается демонстрацией работоспособного приращения функциональности.

Цель — построение MVP (Minimum Viable Product) в течение первого спринта, способного:

- Создавать ТТ в CRM по запросу из GUI или чата

- Перепривязывать ТТ к новому ESR

- Сверять накладные с DMS

- Формировать отчёты в Excel

Последующие спринты направлены на стабилизацию, расширение и интеграцию.

10.2 Стадии разработки

Стадия Цель Продолжительность Результат

----

1. Формирование ТЗ Сбор и формализация требований, утверждение ТЗ 1 неделя Утверждённое ТЗ по ГОСТ 34.602-89

2. Проектирование Разработка архитектуры, POM, структуры БД, GUI 1 неделя Описание программы, схемы, прототипы

3. Реализация (спринты) Поэтапная разработка функциональности 6 недель (3 спринта по 2 недели) Работающая система (MVP + расширения)

4. Тестирование Проверка функциональности, надёжности, безопасности Параллельно со спринтами Отчёт о приёмо-сдаточных испытаниях

5. Внедрение Установка, настройка, обучение пользователя 1 неделя Работающая система в эксплуатации

6. Сопровождение Поддержка, исправление ошибок, доработки Постоянно Обновления, hotfix-релизы

10.3 План спринтов

📆 Спринт 1: MVP (Minimum Viable Product)

Цель: Создать минимально жизнеспособную версию системы, способную выполнять ключевые функции.

Задача ID Тип Описание Ответственный

--

Реализовать dataclass `TradePoint` T1.1 Task Поля: owner, address, region\_code, esr\_code, point\_type и др. Разработчик

Реализовать POM для CRM T1.2 Task Страницы: вход, создание, редактирование ТТ Разработчик

Реализовать POM для DMS T1.3 Task Страницы: вход, выбор подсистемы, поиск накладной Разработчик

Настроить фоновый поток T1.4 Task QThread для запуска процессов без блокировки GUI Разработчик

Реализовать логирование T1.6 Task Логи в `logs/app.log`, уровень DEBUG Разработчик

Реализовать GUI с вкладками T1.11 Task Вкладки: "Создание ТТ", "Сверка", "AI-чат" Разработчик

Реализовать создание ТТ через Selenium US1.2 User Story Автоматизация заполнения формы в CRM Разработчик

Реализовать сверку накладных US2.1–US2.7 User Story Загрузка Excel → фильтрация → проверка в DMS → заливка Разработчик

Реализовать уведомления US1.4 User Story Всплывающее окно: успех/ошибка Разработчик

Реализовать экспорт отчёта US1.13 User Story Excel с GUID, статусом, ошибкой Разработчик

> Критерии приёма MVP:

> - Можно создать ТТ через GUI

> - Можно сверить накладные из Excel

> - Процессы работают в фоне

> - Есть уведомления и логи

> - Приложение не падает при ошибках DMS

📆 Спринт 2: AI-агент и интеллектуальная обработка

Цель: Внедрить AI-агента, обучаемую модель, хранение данных в PostgreSQL.

Задача ID Тип Описание Ответственный

--

Настроить PostgreSQL T1.7 Task Таблицы: `trade\_points`, `region\_esr\_mapping`, `invoice\_reconciliation` Разработчик

Реализовать синхронизацию маппинга T1.8 Task Обновление `region\_code → esr\_code` между Excel и БД Разработчик

Интегрировать LLM (OpenRouter) T1.12 Task Парсинг запросов, генерация JSON Разработчик

Реализовать AI-чат US3.1–US3.3 User Story Диалог: запрос → уточнение → JSON → запуск Аналитик / Разработчик

Реализовать предсказание атрибутов US1.6 User Story Модель предсказывает: `point\_type`, `area`, `location\_type` Разработчик

Реализовать сохранение новых ESR US1.8 User Story При ручном вводе — сохранять в БД Разработчик

Реализовать перепривязку ESR US3.1–US3.3 User Story По XCRM GUID → удалить ESR → добавить новый Разработчик

> Критерии приёма:

> - Можно писать в чат: "Создать ТТ: ООО Ромашка, Новосибирск, ул. Ленина, 15, район Н02"

> - AI формирует JSON и запускает процесс

> - При отсутствии ESR — AI уточняет

> - Новые пары `region\_code → esr\_code` сохраняются в БД

> - Можно перепривязать ТТ к новому ESR

📆 Спринт 3: Стабилизация и подготовка к интеграции с 1С

Цель: Устранить технический долг, улучшить UX, подготовить систему к будущей интеграции.

Задача ID Тип Описание Ответственный

--

Реализовать обработку ошибок DMS T1.14 Task Логирование, продолжение при падении сервера Разработчик

Реализовать валидацию JSON T1.10 Task Проверка обязательных полей перед запуском Разработчик

Реализовать экспорт из PostgreSQL в Excel US4.2 User Story Данные с фильтрами и срезами Аналитик

Добавить таблицу `contractors` — Task Хранение: ИНН, название, текущая сеть Разработчик

Реализовать обогащение `chain\_name` — User Story Использовать `current\_chain` при создании ТТ Разработчик

Подготовить заглушку для 1С — Task Модуль `integration/1c.py` (заготовка) Разработчик

Обновить документацию — Task Руководства, описание программы, ПМИ Аналитик

> Критерии приёма:

> - Система не падает при сетевых ошибках

> - Все процессы валидируются

> - Есть возможность анализировать данные из БД в Excel

> - При смене сети у контрагента — можно пересоздать ТТ

> - Документация актуальна

10.4 Ответственные роли

Роль Ответственность

Аналитик (заказчик) Формулировка требований, приёмка спринтов, тестирование, наполнение данными

Разработчик Реализация модулей, POM, GUI, интеграций, тестирование

QA / Аналитик Проверка соответствия ТЗ, составление ПМИ, фиксация багов

Администратор (опционально) Настройка PostgreSQL, развёртывание на других рабочих станциях

> Примечание:

> На текущем этапе аналитик и разработчик — одно лицо, но роли разделены для соблюдения процесса.

10.5 Критерии приёма системы

Критерий Описание

Функциональность Все user story из спринтов 1–3 реализованы и протестированы

Надёжность Система не падает при ошибках DMS, логирует все операции

Производительность Процессы выполняются в фоне, GUI не блокируется

Безопасность Конфиденциальные данные не логируются, хранятся локально

Документация Все документы (ТЗ, руководства, ПМИ) утверждены

Обучение Аналитик может самостоятельно использовать все функции

10.6 Порядок контроля и приёмки

Этап Действие

Еженедельно Статус-митинг: обсуждение прогресса, блокеров

Конец спринта Демонстрация функциональности аналитику

Приёмка Аналитик подтверждает: "Готово" / "Требует доработки"

После приёмки Фиксация версии в репозитории, обновление документации

Hotfix При критической ошибке — срочный релиз в течение дня

10.7 Выходные документы по стадиям

Стадия Документ

--

Формирование ТЗ Техническое задание (настоящий документ)

Проектирование Описание программы, схемы POM, прототип GUI

Реализация Исходный код, `requirements.txt`, `models/`

Тестирование Программа и методика испытаний (ПМИ), отчёт о приёме

Внедрение Руководство пользователя, руководство администратора

Сопровождение Журнал изменений, список известных проблем