

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

Доцент: Факультет компьютерный наук /  
Департамент программной инженерии,  
ФКН, НИУ ВШЭ

\_\_\_\_\_ Р. А. Нестеров  
«\_\_\_» \_\_\_\_\_ 2025 г.

УТВЕРЖДЕНО

Академический руководитель  
образовательной программы  
«Программная инженерия», старший  
преподаватель департамента  
программной инженерии

\_\_\_\_\_ Н. А. Павловчев  
«\_\_\_» \_\_\_\_\_ 2025 г.

**ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ КОМПЛЕКСНОЙ АВТОМАТИЗАЦИИ  
ПРОЦЕССОВ ПОИСКОВОГО ПРОДВИЖЕНИЯ ВЕБ-САЙТОВ**

**Частное техническое задание**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.13-01 ТЗ 03-1-ЛУ**

Исполнители:

Студент группы БПИ245  
/ А. А. Зобов /

«\_\_\_» \_\_\_\_\_ 2025 г.

Инв. № подп	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
RU.17701729.05.13-01				T3.03_1

УТВЕРЖДЕН

RU.17701729.05.13-01 ТЗ 03-1-ЛУ

**ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ КОМПЛЕКСНОЙ АВТОМАТИЗАЦИИ  
ПРОЦЕССОВ ПОИСКОВОГО ПРОДВИЖЕНИЯ ВЕБ-САЙТОВ**

**Частное техническое задание**

**RU.17701729.05.13-01 ТЗ 03-1**

**Листов 30**

Инв.№ подл	Подп. и дата	Взам. инв.№	Инв.№ дубл.	Подп. и дата
RU.17701729.05.13-01 ТЗ 03-1				

## АННОТАЦИЯ

Настоящее Техническое задание (ТЗ) разработано на создание микросервисного веб-приложение, предназначенного для комплексного управления и оптимизации SEO-продвижения веб-сайтов и является индивидуальной частью общего ТЗ.

Платформа обеспечивает независимость от дорогих внешних SEO API, внедряет механизм Human-in-the-Loop (HITL) для контроля критических изменений и фокусируется на бизнес-ориентированной аналитике, включая расчет метрик FF-Score (Full-Funnel Score) и E-E-A-T Score. Система построена на принципах Domain-Driven Design (DDD) и Event-Driven Architecture (EDA), используя технологический стек Python (FastAPI), React.js и PostgreSQL.

Настоящее ТЗ детализирует требования к функциональным компонентам, относящимся к индивидуальной части разработки (Management Service, API Gateway, Client API Gateway, Frontend и адаптеры), а также формализацию логики оркестрации SEO-процессов и механизма внедрения изменений. Особое внимание уделяется компонентам управления бизнес-логикой, реализации механизма безопасного внедрения изменений через Client API Gateway и разработке пользовательского интерфейса с акцентом на управление оркестрацией задач.

Настоящий документ разработан в соответствии с требованиями:

ГОСТ 19.101-77 [1]: Виды программ и программных документов.

ГОСТ 19.102-77 [2]: Стадии разработки.

ГОСТ 19.103-77 [3]: Обозначения программ и программных документов.

ГОСТ 19.104-78 [4]: Основные надписи.

ГОСТ 19.105-78 [5]: Общие требования к программным документам.

ГОСТ 19.106-78 [6]: Требования к программным документам, выполненным печатным способом.

ГОСТ 19.404-79 [10]: Пояснительная записка. Требования к содержанию и оформлению.

ГОСТ 19.603-78 [12]: Общие правила внесения изменений.

ГОСТ 19.604-78 [13]: Правила внесения изменений в программные документы, выполненные печатным способом.

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [12], ГОСТ 19.604-78 [13].

Иzm.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ .....	5
1.1. Наименование программы .....	5
1.2. Краткая характеристика области применения программы .....	5
2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ .....	6
2.1. Документ(ы), на основании которого(ых) ведется разработка .....	6
2.2. Наименование темы разработки .....	6
3. НАЗНАЧЕНИЕ РАЗРАБОТКИ .....	7
3.1. Функциональное назначение .....	7
3.2. Эксплуатационное назначение .....	7
4. ТРЕБОВАНИЯ К ПРОГРАММЕ .....	8
4.1. Требования к функциональным характеристикам .....	8
4.1.1. Требования к составу выполняемых функций (Микросервисы) .....	8
4.1.2. Схема коммуникации и событийная модель .....	8
4.1.2.1. Требования к сценариям взаимодействия .....	9
4.1.3. Требования к составу выполняемых функций (Management Service) .....	9
4.1.4. Методология: FF-Score (Full-Funnel Score) .....	10
4.1.5. Методология: Human-in-the-Loop (HITL) .....	10
4.1.6. Требования к интеграции: адаптеры и плагины .....	11
4.1.7. Требования к организации входных данных .....	12
4.1.8. Требования к организации выходных данных .....	12
4.1.9. Требования к отчётности .....	13
4.1.10. Формирование PDF и CSV (ориентир по модулям Reporting Service): .....	13
4.1.11. Требования к временным характеристикам .....	13
4.1.12. Требования к интерфейсу .....	13
4.2. Требования к техническому обеспечению .....	14
4.3. Требования к маркировке и упаковке .....	15
4.4. Требования к транспортированию и хранению .....	15
4.5. Требования к программному обеспечению .....	15
4.6. Требования к информационной безопасности .....	15
4.7. Требования к надежности .....	15
4.8. Требования к сопровождению .....	16

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4.9. Специальные требования .....	16
<b>5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ .....</b>	<b>17</b>
5.1. Предварительный состав программной документации .....	17
5.2. Требования к программной документации .....	17
<b>6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ .....</b>	<b>18</b>
6.1. Предполагаемая потребность .....	18
6.2. Целевая аудитория .....	18
6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами .....	18
<b>7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ .....</b>	<b>22</b>
7.1. Стадии и этапы разработки .....	22
7.2. Сроки разработки и исполнители .....	24
<b>8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ .....</b>	<b>26</b>
8.1. Виды испытаний .....	26
8.2. Общие требования к приёмке работы .....	26
<b>ПРИЛОЖЕНИЕ 1 .....</b>	<b>27</b>
<b>ПРИЛОЖЕНИЕ 2 .....</b>	<b>30</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 1. ВВЕДЕНИЕ

### **1.1. Наименование программы**

Наименование программы – «Веб-приложение для комплексной автоматизации процессов поискового продвижения веб-сайтов».

Наименование программы на английском языке – «Web-application for the comprehensive automation of website search engine optimization processes».

### **1.2. Краткая характеристика области применения программы**

Веб-приложение предназначено для комплексного управления, автоматизации и оптимизации SEO-продвижения веб-сайтов. Область применения охватывает технический аудит, семантический анализ, генерацию контента, управление задачами и бизнес-ориентированную отчетность. Программа нацелена на автоматизацию и повышение эффективности SEO-процессов. Одной из уникальных черт является использование собственной метрики FF-Score (Full-Funnel Score), которая напрямую связывает технические и контентные показатели с воронкой продаж и конверсиями.

В настоящее время рынок SEO продвижения сталкивается с рядом критических проблем, которые не решаются существующими инструментами:

- 1. Высокая стоимость внешних API:** Дорогие подписки на специализированные сервисы вроде, что делает SEO-оптимизацию дорогостоящей для разработки для нужд малого и среднего бизнеса.
- 2. Отсутствие Контроля над Автоматизацией:** Существующие инструменты либо полностью ручные (Screaming Frog), либо автоматизированы без возможности контроля критических изменений, что приводит к риску SEO-катастроф.
- 3. Фокус на Технических Метриках:** Большинство инструментов фокусируются на технических показателях (DR, DA, ошибки краулинга) [19], игнорируя прямую связь с бизнес-результатами (конверсии, выручка).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

### 2.1. Документ(ы), на основании которого(ых) ведется разработка

Разработка ведётся на основании учебного плана (ГОСТ 19.201–78 — ТЗ)[7] подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденной академическим руководителем программы темы курсового проекта.

### 2.2. Наименование темы разработки

Наименование темы разработки: Веб-приложение для комплексной автоматизации процессов поискового продвижения веб-сайтов.

Наименование темы разработки на английском языке – «Web-application for the comprehensive automation of website search engine optimization processes».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### **3. НАЗНАЧЕНИЕ РАЗРАБОТКИ**

#### **3.1. Функциональное назначение**

Разрабатываемая программа предназначена для:

- **Оркестрации SEO-задач** (SEO Robot) на основе метрики FF-Score.
- **Безопасного внедрения изменений** на сайт клиента через Client API Gateway и HITL-контроль.
- **Бизнес-ориентированной отчетности** (Cost-Efficiency, HITL-Efficiency).

#### **3.2. Эксплуатационное назначение**

Основными конечными потребителями являются SEO-специалисты, владельцы бизнеса и маркетинговые команды, которым требуется:

- Снизить операционные расходы на внешние SEO-инструменты.
- Получить прозрачный и контролируемый процесс внедрения SEO-изменений.
- Фокусироваться на стратегических задачах, делегируя рутину автоматизированному «SEO Robot».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 4. ТРЕБОВАНИЯ К ПРОГРАММЕ

### 4.1. Требования к функциональным характеристикам

#### 4.1.1. Требования к составу выполняемых функций (Микросервисы)

Программа должна быть реализована как набор микросервисов, каждый из которых выполняет определенный ограниченный контекст (Bounded Context) [2].

*Таблица 1*

Компоненты системы и их ключевые функции

Микросервис	Назначение	Ключевые Функции
<b>Management Service</b>	Оркестратор бизнес-логики (SEO Robot).	Запуск циклов оптимизации, приоритизация задач (Impact x Effort), управление очередью HTML-одобрений.
<b>Frontend (SPA)</b>	Пользовательский интерфейс для управления SEO-процессами и HTML-контроля	Аутентификация и управление профилем, визуализация FF-Score в дашборде, интерфейс для HTML-одобрений с DiffViewer (сравнение «до/после»), просмотр и экспорт отчетов, адаптивный дизайн для устройств от 768px
<b>Client API Gateway</b>	Безопасный шлюз для внедрения изменений.	Аутентификация (HMAC-SHA256), ограниченные PATCH эндпоинты, логирование изменений.

#### 4.1.2. Схема коммуникации и событийная модель

Система использует событийную модель для согласования работы сервисов и минимизации жестких связей. Ключевые изменения жизненного цикла задач публикуются как доменные события. Frontend является активным потребителем событий для обеспечения актуального состояния пользовательского интерфейса в режиме реального времени.

Иzm.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## Обязательные доменные события и их назначение

Событие	Источник	Потребители	Назначение
<b>TaskCreated</b>	Management Service	Audit Service, Semantic Service, Frontend	Запуск рабочего процесса и отображение новой задачи в интерфейсе
<b>CrawlCompleted</b>	Audit Service	Management Service, Semantic Service	Уведомление о готовности сырых данных.
<b>FFScoreRecalculated</b>	Semantic Service	Management Service, Reporting Service, Frontend	Триггер для переприоритизации задач и обновления отчётов.
<b>HITLApproved</b>	API Gateway	Management Service, Frontend	Разрешение на внедрение изменений на сайт клиента и обновление состояния в интерфейсе

**4.1.2.1. Требования к сценариям взаимодействия**

Основные сценарии взаимодействия между компонентами системы с фокусом на оркестрацию:

**TaskCreated**: после постановки задачи Management Service публикует событие;

**Audit** и **Semantic** запускают сбор/анализ данных, Frontend обновляет список активных задач.

**Оркестрация SEO-циклов**: Management Service периодически запускает run\_optimization\_cycle(project\_id), который анализирует текущий FF-Score и выбирает соответствующий режим работы («Спасение» при FF-Score < 40 или «Рост» при FF-Score > 60).

**Приоритизация задач**: Management Service применяет алгоритм prioritize\_tasks(task\_list) по формуле Impact × Effort для определения порядка выполнения задач и формирует очередь для обработки.

**HITLApproved**: после ручного одобрения в UI API Gateway публикует событие;

Management Service потребляет событие, вызывает deploy\_changes через Client API Gateway для внедрения изменений на стороне клиента, Frontend обновляет состояние задачи в интерфейсе.

**4.1.3. Требования к составу выполняемых функций (Management Service)**

В составе Management Service должны быть реализованы следующие ключевые компоненты:

- tasks/celery\_app.py — конфигурация Celery и базовые задачи.
- tasks/periodic\_tasks.py — daily\_ff\_score\_recalculation() (Celery Beat).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- tasks/orchestrator.py — run\_optimization\_cycle(project\_id) и переприоритизация prioritize\_tasks(task\_list) по Impact × Effort:

При FF-Score < 40 активируется режим «Спасение» с приоритетом на исправление критических ошибок (CWV, 404, robots.txt, Schema.org).

При FF-Score > 60 активируется режим «Рост» с приоритетом на стратегические улучшения (Content Gap анализ, Е-Е-А-Т оптимизация, внутренняя перелинковка).

При  $40 \leq \text{FF-Score} \leq 60$  применяется сбалансированный режим с равномерным распределением задач по всем категориям.

- client\_api\_adapter.py — безопасное взаимодействие с Client API Gateway; deploy\_changes(task\_id, changes\_data) после прохождения HITL
- db.py — модели БД для tasks, hitl\_approvals, changelog.
- interlink\_generator.py - модуль для генерации внутренней перелинковки по правилам: анализ контентной релевантности, распределение веса страниц, предотвращение перелинковки нерелевантных разделов. Генерирует предложения по добавлению внутренних ссылок с указанием anchor-текста и целевой страницы.

#### 4.1.4. Методология: FF-Score (Full-Funnel Score)

FF-Score рассчитывается в Semantic Service и является агрегированным KPI пользователя, а также триггером для Management Service (переприоритизация, выбор режима оптимизации).

Таблица 3

#### Компоненты FF-Score и примерные веса

Компонент	Входные данные	Вес (примерный)
<b>Freshness</b>	Давность апдейта, сдвиг SERP, LinkVelocity (из GSC [17]/Собственного Индекса)	30%
<b>Familiarity / Semantic Distance [14]</b>	Покрытие СЯ, Е-Е-А-Т сигналы [14], семантическое расстояние [18] от ТОП-10 (из Semantic Service)	40%
<b>Quality</b>	CWV, 404, Schema.org [14] валидация (из Audit Service)	30%

#### 4.1.5. Методология: Human-in-the-Loop (HITL)

HITL — обязательный механизм безопасности для критических изменений (Title/ Description/

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

H1/Schema.org [14], правки контента и др.). HITL реализуется в Management Service (логика и хранение) и API Gateway/Frontend (интерфейс просмотра/одобрения).

Последовательность процесса:

- Детектирование: Management Service определяет необходимость изменения и оценивает Impact (потенциал влияния на FF-Score).
- Генерация черновика: Semantic Service генерирует черновик изменения (например, новый Title/Description/H1/Schema.org [14]).
- Создание записи на одобрение: Management Service создаёт запись в tasks и/или hitl\_approvals с diff\_data (JSON «до/после»), impact\_score, временными метками.
- Ручная проверка: в HITL-UI пользователь видит DiffViewer (до/после) и принимает решение approve/reject.
- Публикация события: при одобрении API Gateway публикует событие HITLApproved.
- Внедрение и логирование: Management Service потребляет HITLApproved, вызывает deploy\_changes через client\_api\_adapter.py и фиксирует запись в changelog.

Минимальные требования к сущностям хранения:

- tasks: идентификатор, проект, статус, ссылки на результаты аудита/семантики, impact\_score.
- hitl\_approvals: связь с задачей, diff\_data, статус, approved\_by, approved\_at.
- changelog: связь с задачей/изменением, что внедлено, когда, кем/как (auto или HITL).

#### **4.1.6. Требования к интеграции: адаптеры и плагины**

Для внедрения изменений на стороне клиента должны быть предоставлены следующие адAPTERЫ/плагины:

##### **WordPress Plugin:**

Совместимость с последними 3 версиями WordPress (6.4+) Совместимость с популярными SEO-плагинами (Yoast SEO, RankMath) в режиме «перехвата» и приоритета выполнения

##### **Структура файлов:**

seo-master-plugin.php — основной файл плагина, определение хуков и загрузка компонентов

admin/settings.php — страница настроек в админке WordPress (ввод API-токена, выбор режима работы)

includes/meta-filter.php — хуки перехвата и замены Title/Description/Schema.org на уровне вывода

includes/api-handler.php — обработчик запросов от Client API Gateway

uninstall.php — корректное удаление данных при деактивации плагина

##### **Универсальный PHP-адаптер:**

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Реализация в виде одного файла с минимальными зависимостями

Использование PDO для безопасного взаимодействия с БД с защитой от SQL-инъекций:

- Все запросы должны использовать параметризованные выражения
- Минимальные права доступа к БД (только необходимые таблицы)
- Валидация типов данных перед выполнением операций

Структура файлов:

- adapter.php — легковесный API-адаптер для самописных сайтов
- db\_patcher.php — безопасное обновление записей в локальной БД клиента с использованием PDO-транзакций

#### **Python-адаптер:**

Middleware для интеграции в проекты на Django/Flask/FastAPI:

- Для Django: наследование от django.middleware.common.MiddlewareMixin
- Для Flask: использование декораторов @app.before\_request и @app.after\_request
- Для FastAPI: интеграция через middleware и Depends

Поддержка асинхронных операций для FastAPI/Starlette

Структура файлов:

- adapter.py — базовый адаптер с функцией инициализации
- middlewares/django\_middleware.py — специализированный middleware для Django
- middlewares/flask\_middleware.py — специализированный middleware для Flask
- middlewares/fastapi\_middleware.py — специализированный middleware для FastAPI

#### **4.1.7. Требования к организации входных данных**

Входные данные для системы делятся на три категории:

- **Пользовательский Ввод (Frontend):** URL проекта, API-ключи GSC/GA4 [16] [17], подтверждение HTML-одобрения.
- **Внешние API:** Данные, получаемые через API (GSC, GA4, PSI, LLM-провайдеры).
- **Сырые Данные:** HTML-код, CSS, JS, данные о ссылках, собранные собственным краулером и индексатором.

#### **4.1.8. Требования к организации выходных данных**

Выходные данные должны быть представлены в следующих формах:

- **Пользовательский Интерфейс (Frontend):** Визуализация FF-Score, списки задач, детальные карточки HTML-одобрений с DiffViewer.
- **Отчеты:** PDF-отчеты (еженедельные, ежемесячные) и CSV-экспорт сырых данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- **Изменения на Сайте Клиента:** Безопасные PATCH-операции, внедренные через Client API Gateway.

#### **4.1.9. Требования к отчётности**

- Оперативный отчет (еженедельный): позиции по кластерам, CTR, CWV (доля Good), ошибки сканирования, статус HITL-одобрений.
- Стратегический отчет (ежемесячный): FF-Score, лиды/выручка из органики, Cost-Efficiency, HITL-Efficiency.
- Changelog: журнал всех внедренных изменений (автоматических и HITL-одобренных) для связи метрик с изменениями.

#### **4.1.10. Формирование PDF и CSV (ориентир по модулям Reporting Service):**

- pdf\_generator.py — функция generate\_weekly\_report(); генерация PDF на основе шаблонов (например, Jinja2) с таблицами и графиками (например matplotlib/plotly).
- csv\_exporter.py — функция export\_raw\_data(); экспорт сырых данных в CSV с корректной кодировкой UTF-8.
- Сборщики данных (минимум): gsc\_collector.py [17], ga4\_collector.py [16], social\_collector.py yandex\_webmaster\_collector.py.
- reporting\_service/metrics/calculator.py — расчёт дополнительных показателей (например, Trust/Sentiment, HITL-Efficiency)

- Изменения на Сайте Клиента: Безопасные PATCH-операции, внедренные через Client API Gateway

#### **4.1.11. Требования к временным характеристикам**

- **Синхронные Запросы (API Gateway):** Время ответа не должно превышать 500 мс.
- **Асинхронные Задачи (Celery):**

- Полный технический аудит: не более 24 часов для сайта до 100 000 страниц.
- Расчет FF-Score: не более 1 часа.
- LLM-генерация черновика: не более 5 минут.

#### **4.1.12. Требования к интерфейсу**

Интерфейс должен быть реализован как Single Page Application (SPA) с использованием React/Vue.js.

##### **1. Общие требования**

1.1. Адаптивность: Интерфейс должен корректно отображаться на дисплеях с разрешением от 1280px (Desktop) до 768px (Tablet).

1.2. Техническая эстетика: Дизайн должен соответствовать современным гайдлайнам (на-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

пример, Material Design) и обеспечивать высокую читаемость.

1.3. Производительность: Время загрузки основного интерфейса (LCP) не должно превышать 2.5 секунды. [19]

## 2. Раздел «Авторизация и Профиль»

2.1. Окно входа: Должно содержать поля для ввода логина и пароля, а также кнопки «Войти» и «Забыли пароль».

2.2. Профиль: Должен предоставлять возможность ввода и сохранения API-ключей GSC/GA4, а также кнопку «Выход».

## 3. Раздел «Дашборд (Обзор проекта)»

3.1. Визуализация FF-Score: Главный элемент, отображающий текущий FF-Score в виде графика с историей изменений и числовым значением.

3.2. Список активных задач: Виджет, отображающий топ-5 приоритизированных задач, ожидающих выполнения или одобрения.

3.3. Управление аудитом: Кнопка «Запустить новый аудит» с возможностью выбора типа аудита (технический, контентный).

3.4. Статистика: Виджеты для отображения Cost-Efficiency и HITL-Efficiency.

## 4. Раздел «HITL-контроль (Одобрение изменений)»

4.1. Список изменений: Таблица с фильтрацией и сортировкой, отображающая все предложенные изменения с полями: Тип изменения, Приоритет FF-Score, Дата создания, Статус.

4.2. Детальный просмотр: Должен включать DiffViewer для визуального сравнения старого и нового кода/текста (например, мета-тегов, контента).

4.3. Управление: Кнопки «Одобрить», «Отклонить», «Редактировать» с обязательным полем для комментария при отклонении.

## 5. Технические требования к компонентам

5.1. DiffViewer: Реализовать компонент на основе библиотеки react-diff-viewer с поддержкой синтаксической подсветки для HTML, JSON и текста. Поддерживать режимы «Страна-к-стороне» и «Объединенный».

5.2. Роутинг: Использовать React Router v6 для навигации между страницами с сохранением состояния при перезагрузке.

5.3. Состояние: Использовать Redux Toolkit для управления глобальным состоянием приложения, включая аутентификацию, задачи и события.

## 4.2. Требования к техническому обеспечению

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Разработка и эксплуатация должны быть возможны на стандартном серверном оборудовании с поддержкой контейнеризации (Docker).

- **Процессор:** x86-64, не менее 4 ядер.
- **Оперативная память:** Не менее 16 ГБ (для Production-среды).
- **Дисковое пространство:** SSD, не менее 500 ГБ (для хранения индекса ссылок и данных краулинга).

#### **4.3. Требования к маркировке и упаковке**

Программный продукт поставляется в виде электронного комплекта, включающего программную документацию, само приложение (исполнимые файлы, сопутствующие рабочие файлы, а также исходный код) и презентационные материалы проекта. Требования к маркировке и упаковке отсутствуют..

#### **4.4. Требования к транспортированию и хранению**

Программный продукт должен транспортироваться таким образом, чтобы сохранялась полнота исходного комплекта, предоставленного разработчиком.

Хранение продукта организовано в репозитории на платформе GitHub.

#### **4.5. Требования к программному обеспечению**

*Таблица 4*

##### Требования к программному обеспечению

Компонент	Требование
<b>Операционная Система</b>	Linux (Ubuntu/Debian)
<b>Backend</b>	Python 3.10+, FastAPI
<b>База Данных</b>	PostgreSQL 14+ (с поддержкой JSONB)
<b>Очередь Задач</b>	Celery, Redis/RabbitMQ
<b>Контейнеризация</b>	Docker, Docker Compose
<b>Краулинг</b>	Playwright (для JS-рендеринга), Scrapy

#### **4.6. Требования к информационной безопасности**

- **Аутентификация:** Использование JWT-токенов для API Gateway.
- **Безопасность Клиента:** Client API Gateway должен использовать HMAC-SHA256 для подписи запросов и принимать только JSON-Patch формат для минимизации рисков [2].
- **Разграничение Доступа:** Реализация ролевой модели (Администратор, Пользователь, Гость).
- **Хранение Секретов:** Использование переменных окружения или Vault для хранения API-ключей.

#### **4.7. Требования к надежности**

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- **Отказоустойчивость:** Использование Docker Swarm/Kubernetes для Production-развертывания с автоматическим перезапуском контейнеров.

- **Атомарность:** Использование транзакций в Client API Gateway для обеспечения атомарности операций записи в БД клиента.

- **Логирование:** Централизованное логирование всех операций и ошибок.

#### 4.8. Требования к сопровождению

- **Документация:** Подробная документация по API (OpenAPI/Swagger) для всех микросервисов.

- **Мониторинг:** Настройка Prometheus и Grafana для сбора и визуализации метрик [2].

- **Тестирование:** Покрытие кода Unit- и Integration-тестами.

#### 4.9. Специальные требования

Особые требования к программе отсутствуют.

Иzm.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

### 5.1. Предварительный состав программной документации

- Техническое Задание (настоящий документ).
- Архитектурное Описание Системы.
- Руководство Программиста (API-документация).
- Руководство Пользователя.
- Программа и Методика Испытаний.

### 5.2. Требования к программной документации

Документация должна быть выполнена в соответствии с требованиями Единой системы программной документации (ЕСПД) [1–13] и включать все необходимые разделы для сопровождения и развития проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

### 6.1. Предполагаемая потребность

На современном рынке SEO-инструментов наблюдается ряд системных проблем, препятствующих эффективному и безопасному продвижению веб-сайтов. К ним относятся: высокая стоимость коммерческих решений, отсутствие контроля над автоматически внедряемыми изменениями, а также фокус исключительно на технических параметрах без учёта бизнес-результатов (конверсий, выручки).

Разрабатываемое веб-приложение призвано устранить указанные недостатки за счёт реализации трёх ключевых принципов:

1. Независимости от внешних API-сервисов (собственный краулер и индекс ссылок);
2. Контролируемой автоматизации (механизм Human-in-the-Loop);
3. Бизнес-ориентированной аналитики (метрика FF-Score, интеграция с GA4/Яндекс.Метрикой).

### 6.2. Целевая аудитория

Основными пользователями программного продукта являются:

1. SEO-специалисты, стремящиеся снизить зависимость от сторонних платных инструментов и повысить рентабельность своей работы за счёт автоматизации рутинных операций;
2. Маркетологи и владельцы бизнеса, заинтересованные в прозрачной связи между SEO-действиями и ключевыми бизнес-показателями (лиды, продажи, ROI);
3. Разработчики веб-сайтов, которым необходимы инструменты для объективной оценки технического состояния ресурсов (Core Web Vitals, Mobile-First, Schema.org [14] [19]).

### 6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Для обоснования экономической целесообразности разработки проведено сравнительное исследование с наиболее распространенными российскими и зарубежными SEO-платформами. Анализ охватывает 25 функциональных критериев, сгруппированных по пяти категориям: технический аудит, семантика и контент, ссылочный профиль, автоматизация, бизнес-аналитика.

Результаты сравнения представлены в таблицах 1 и 2. Для унификации оценки принята следующая система обозначений:

- «+» — функция полностью реализована;
- «±» — функция реализована частично или с ограничениями;
- «-» — функция отсутствует.

Иzm.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 5

## Сравнение функциональных характеристик с зарубежными SEO-аналогами

Функция	Ahrefs [25]	Semrush [26]	Screaming Frog [27]	PromoPult [28]	Rookeee [29]	Наша раз- работка
JS/SPA-рендеринг	-	-	+	-	-	+
Core Web Vitals монито- ринг	-	-	-	-	-	+
E-E-A-T Score [14]	-	-	-	-	-	+
FF-Score (SEO ↔ бизнес)	-	-	-	-	-	+
HITL-контроль измене- ний	-	-	-	-	-	+
LLM-генерация контента	±	±	-	±	-	+
Собственный индекс ссы- лок	+	+	-	-	+	+
Безопасное внедрение че- рез API	-	-	-	-	-	+
Cost-Efficiency / HITL- Efficiency	-	-	-	-	-	+
Микросервисная архи- тектура	-	-	-	-	-	+
<b>Итого</b>	<b>1.5</b>	<b>1.5</b>	<b>1</b>	<b>0.5</b>	<b>1</b>	<b>10</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 6

## Сравнение функциональных характеристик с российскими SEO-аналогами

Функция	MegaIndex [30]	WebEffecto[r] [31]	Key Collector [32]	Netpeak Spider [33]	Rush Analytics [34]	Наша разработка
JS/SPA-рендеринг	-	-	-	-	-	+
Core Web Vitals интеграция	-	-	-	±	+	+
HMTL-контроль	-	±	-	-	-	+
LLM-генерация контента	-	-	-	-	-	+
E-E-A-T Score [14]	-	-	-	-	-	+
FF-Score	-	-	-	-	-	+
Плагин для WordPress с API	-	±	-	-	±	+
Микросервисная архитектура	-	-	-	-	-	+
<b>Итого</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0.5</b>	<b>1.5</b>	<b>8</b>

Согласно проведённому сравнению, «SEO Мастер» превосходит все рассмотренные аналоги по совокупности функциональных возможностей, обеспечивая максимальное покрытие современных требований к SEO-оптимизации.

Основные экономические эффекты от внедрения программного продукта:

1. Снижение операционных расходов (OpEx) на SEO-инструменты — до 70–90% по сравнению с

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

подписками на Ahrefs/Semrush;

2. Повышение рентабельности SEO-кампаний за счёт перехода от технических метрик к бизнес-KPI (Cost-Efficiency, ROI) [16];
3. Сокращение трудозатрат SEO-специалистов на рутинные операции (аудит, сбор семантики, генерация ТЗ) за счёт автоматизации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

### 7.1. Стадии и этапы разработки

Стадии и этапы разработки были выявлены с учётом ГОСТ 19.102-77 [2].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 7

## Стадии и этапы разработки

Стадия разработки	Этап работ	Содержание работ	Исполнители работ	Сроки выполнения
Техническое задание	Обоснование необходимости разработки	Постановка задачи	Зобов Арсений Александрович	20.10.25
	Научно-исследовательский этап разработки	Определение структуры входных и выходных данных	Зобов Арсений Александрович	23.10.25 – 15.12.25
		Определение требований к техническим и программным средствам	Зобов Арсений Александрович	23.10.25 – 15.12.25
	Разработка и утверждение технического задания	Определение требований к программному продукту	Зобов Арсений Александрович	23.10.25 – 15.12.25
Рабочий проект	Проектирование архитектуры	Разработка моделей данных и схемы событийной шины	Зобов Арсений Александрович	17.12.25 – 10.01.25
		Формализация логики HTML-механизма	Зобов Арсений Александрович	17.12.25 – 10.01.25
	Разработка программы	Реализация Management Service и API Gateway	Зобов Арсений Александрович	11.01.26 – 25.02.26
		Реализация адаптеров для клиентов	Зобов Арсений Александрович	01.03.26 – 25.03.26

Продолжение табл. 7

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Стадия разработки	Этап работ	Содержание работ	Исполнители работ	Сроки выполнения
		Разработка Frontend (SPA на React, HITL-UI с DiffViewer)	Зобов Арсений Александрович	01.02.26 – 20.03.26
	Разработка программной документации	Разработка документов в соответствии с требованиями ГОСТ 19 ЕСПД (Единой системы программной документации)	Зобов Арсений Александрович	20.10.25 – 05.04.26
Испытания программы	Испытания программы	Разработка, согласование и утверждение программы и методики испытаний	Зобов Арсений Александрович	21.03.26 – 05.04.26
		Корректировка программы и программной документации по результатам испытаний	Зобов Арсений Александрович	16.04.26 – 20.04.26
Внедрение	Подготовка и передача программы	Подготовка программы и программной документации для презентации и защиты	Зобов Арсений Александрович	21.04.26 – 25.04.26
		Загрузка материалов курсового проекта в ЛМС, дисциплина «Курсовой проект, 2 курс, ПИ»	Зобов Арсений Александрович	xx.04.26
		Защита программного продукта комиссии	Зобов Арсений Александрович	xx.04.26

,

## 7.2. Сроки разработки и исполнители

Разработка программного продукта должна быть завершена не позже утверждённого срока сдачи работы.

Иzm.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Исполнитель – Зобов Арсений Александрович, студент группы БПИ245 факультета компьютерных наук НИУ ВШЭ.

Иzm.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Контроль и приемка разработки осуществляются в соответствии с документом «Программа и методика испытаний» (ГОСТ 19.301–79 [8]).

### 8.1. Виды испытаний

1. **Функциональное Тестирование:** Проверка корректности работы всех микросервисов и их взаимодействия.
2. **Нагрузочное Тестирование:** Проверка стабильности системы при пиковых нагрузках (например, при массовом краулинге).
3. **Приемочные Испытания:** Проверка соответствия функционала требованиям ТЗ, включая корректность расчета FF-Score и работу HITL-механизма.

### 8.2. Общие требования к приёмке работы

Программа считается принятой при условии:

1. Корректной работы всех функций, описанных в разделе 4.1.1.
2. Успешного прохождения приемочных испытаний.
3. Предоставления полного комплекта программной документации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 1****СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.101-77: Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77: Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77: Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78: Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78: Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78: Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78: Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.301-79: Программа и методика испытаний. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.401-78: Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.404-79: Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 19.505-79: Руководство оператора. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. ГОСТ 19.603-78: Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. ГОСТ 19.604-78: Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

14. Google Search Essentials. Электронный ресурс. URL: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide> (дата обращения: 14.12.2025)
15. OpenAI API Documentation: Prompt Engineering. Электронный ресурс. URL: <https://platform.openai.com/docs/guides/prompt-engineering> (дата обращения: 14.12.2025)
16. Google Analytics 4 Reporting API. Электронный ресурс. URL: <https://developers.google.com/analytics/devguides/reporting/data/v1> (дата обращения: 14.12.2025)
17. Google Search Console API Documentation. Электронный ресурс. URL: <https://developers.google.com/webmaster-tools/search-console-api-original/v3> (дата обращения: 14.12.2025)
18. Jurafsky, D., Martin, J. H. Speech and Language Processing. — 3rd ed. — Stanford University, 2023. Электронный ресурс. URL: <https://web.stanford.edu/~jurafsky/slp3/> (дата обращения: 14.12.2025)
19. Google – Core Web Vitals. Электронный ресурс. URL: <https://web.dev/vitals/> (дата обращения: 14.12.2025)
20. Playwright Documentation. Электронный ресурс. URL: <https://playwright.dev/> (дата обращения: 14.12.2025)
21. Scrapy Documentation. Электронный ресурс. URL: <https://docs.scrapy.org/> (дата обращения: 14.12.2025)
22. Celery Documentation. Электронный ресурс. URL: <https://docs.celeryq.dev/> (дата обращения: 14.12.2025)
23. FastAPI Documentation. Электронный ресурс. URL: <https://fastapi.tiangolo.com/> (дата обращения: 14.12.2025)
24. React Documentation. Электронный ресурс. URL: <https://react.dev/> (дата обращения: 14.12.2025)
25. Ahrefs: [Электронный ресурс]. URL: <https://ahrefs.com> (дата обращения: 14.12.2025 ).
26. Semrush: [Электронный ресурс]. URL: <https://semrush.com> (дата обращения: 14.12.2025 ).
27. Screaming Frog SEO Spider: [Электронный ресурс]. URL: <https://www.screamingfrog.co.uk/seo-spider/> (дата обращения: 14.12.2025 ).
28. PromoPult: [Электронный ресурс]. URL: <https://promopult.ru/technology> (дата обращения: 14.12.2025 ).
29. Rookee: [Электронный ресурс]. URL: <https://rookee.ru/vitrina-uslug/seo/> (дата обращения: 14.12.2025 ).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

30. MegaIndex: [Электронный ресурс]. URL: <https://ru.megaindex.com/> (дата обращения: 14.12.2025 ).
31. WebEffect: [Электронный ресурс]. URL: <https://www.webeffect.ru/> (дата обращения: 14.12.2025 ).
32. Key Collector: [Электронный ресурс]. URL: <https://keycollector.ru/> (дата обращения: 14.12.2025 ).
33. Netpeak Spider: [Электронный ресурс]. URL: <https://netpeak.net/ru/spider/> (дата обращения: 14.12.2025 ).
34. Rush Analytics: [Электронный ресурс]. URL: <https://rush-analytics.com/> (дата обращения: 14.12.2025 ).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 Т3 03-1				
Инв. № подл.				
	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 2****ТЕРМИНОЛОГИЯ***Таблица 2.1*

Термин	Определение
API Gateway	Единая точка входа для всех клиентов, управляющая маршрутизацией, безопасностью и Rate Limiting.
DDD (Domain-Driven Design)	Подход к разработке ПО, фокусирующийся на основном домене и его логике.
EDA (Event-Driven Architecture)	Архитектурный паттерн, в котором взаимодействие компонентов происходит через события.
E-E-A-T Score	Метрика качества контента (Experience, Expertise, Authoritativeness, Trustworthiness) в SEO.
FF-Score (Full-Funnel Score)	Агрегированная метрика (0-100) для оценки эффективности SEO-продвижения.
HITL (Human-in-the-Loop)	Механизм, требующий человеческого одобрения для критических автоматизированных изменений.
LLM (Large Language Model)	Большая языковая модель, используемая для генерации контента и семантического анализа.
Микросервис	Слабосвязанный, независимо развертываемый сервис, реализующий определенный ограниченный контекст.
SEO (Search Engine Optimization)	Комплекс мер по оптимизации веб-сайта для улучшения его позиций в поисковых системах.
SPA (Single Page Application)	Веб-приложение, загружающее все ресурсы на одной странице и динамически обновляющее контент.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-01 ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

## Лист регистрации изменений