

**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 компьютерных наук /
Департамент программной инженерии,
ФКН, НИУ ВШЭ

_____ Р. А. Нестеров
«__» _____ 2026 г.

УТВЕРЖДЕНО

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

_____ Н. А. Павлочев
«__» _____ 2026 г.

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

Частное техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.13-02.ТЗ 03-1-ЛУ

Исполнители:
Студент группы БПИ245

_____ / А. А. Зобов /

«__» _____ 2026 г.

Инв.№ подл RU.17701729.05.13-02.ТЗ 03-1	Подп. и дата	Взам. инв.№	Инв.№ дубл.	Подп. и дата

УТВЕРЖДЕН

RU.17701729.05.13-02.ТЗ 03-1-ЛУ

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

Частное техническое задание

RU.17701729.05.13-02.ТЗ 03-1

Листов 33

Инов.№ подл	Подп. и дата	Взам. инв.№	Инов.№ дубл.	Подп. и дата
RU.17701729.05.13-02.ТЗ 03-1				

АННОТАЦИЯ

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

Платформа использует интеграции с Google Search Console и Яндекс Метрку, 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 специализированные адаптеры для WordPress и Tilda), 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].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.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	9
4.1.2.1. Требования к сценариям взаимодействия	9
4.1.3. Требования к составу выполняемых функций (Management Service)	10
4.1.4. Методология: FF-Score (Full-Funnel Score)	10
4.1.5. Методология: Human-in-the-Loop (HITL)	11
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	12
4.1.10. Формирование CSV (ориентир по модулям Reporting Service):	13
4.1.11. Требования к временным характеристикам	13
4.1.12. Требования к интерфейсу	13
4.1.12.1. Публичный интерфейс первичного аудита	14
4.2. Требования к техническому обеспечению	15
4.3. Требования к маркировке и упаковке	15
4.4. Требования к транспортированию и хранению	15
4.5. Требования к программному обеспечению	16
4.6. Требования к информационной безопасности	16
4.6.1. Детализация НМАС-аутентификации	16

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4.7. Требования к отказоустойчивости и мониторингу	17
4.8. Требования к сопровождению	17
4.9. Специальные требования	17
5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ	18
5.1. Предварительный состав программной документации	18
5.2. Требования к программной документации	18
6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	19
6.1. Предполагаемая потребность	19
6.2. Целевая аудитория	19
6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами	19
6.3.1. Ограничения и риски разрабатываемой системы	23
7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ	24
7.1. Стадии и этапы разработки	24
7.2. Сроки разработки и исполнители	26
8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ	28
8.1. Виды испытаний	28
8.2. Общие требования к приёмке работы	28
ПРИЛОЖЕНИЕ 1	29
ПРИЛОЖЕНИЕ 2	33

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 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.17701729.05.13-02.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.17701729.05.13-02.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.17701729.05.13-02.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
Management Service	Оркестратор бизнес-логики (SEO Robot).	Запуск циклов оптимизации, приоритизация задач (Impact x Effort), управление очередью HITL-одобрений.
Frontend (SPA)	Пользовательский интерфейс для управления SEO-процессами и HITL-контроля	Аутентификация и управление профилем, визуализация FF-Score в дашборде, интерфейс для HITL-одобрений с DiffViewer (сравнение «до/после»), просмотр и экспорт отчетов в CSV, адаптивный дизайн для устройств от 768px
Client API Gateway	Безопасный шлюз для внедрения изменений.	Аутентификация (HMAC-SHA256), ограниченные PATCH эндпоинты, логирование изменений.

Публичные эндпоинты (без аутентификации):

POST /api/public/quick-audit

- Body: {«url»: «https://example.com»}
- Rate limit: 5 req/hour per IP
- Response: {«audit_id»: «uuid», «status»: «processing»}

GET /api/public/audit-status/{audit_id}

- Polling эндпоинт для проверки статуса
- Response:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- status: «processing» | «completed» | «failed»
- progress: 0-100
- result: {...} (при status=«completed»)

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.

Таблица 2

Обязательные доменные события и их назначение

Событие	Источник	Потребители	Назначение
TaskCreated	Management Service	Audit Service, Semantic Service, Frontend	Запуск рабочего процесса и отображение новой задачи в интерфейсе
CrawlCompleted	Audit Service	Management Service, Semantic Service	Уведомление о готовности сырых данных.
FFScoreRecalculated	Semantic Service	Management Service, Reporting Service, Frontend	Триггер для переприоритизации задач и обновления отчётов.
HITLApproved	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).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Приоритизация задач: Management Service применяет алгоритм `prioritize_tasks(task_list)` по формуле $\text{Impact} \times \text{Effort}$ 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).
- `tasks/orchestrator.py` — `run_optimization_cycle(project_id)` и переприоритизация `prioritize_tasks(task_list)` по $\text{Impact} \times \text{Effort}$:

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

При $\text{FF-Score} > 60$ активируется режим «Рост» с приоритетом на стратегические улучшения (Content Gap анализ, E-E-A-T 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).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 3

Компоненты FF-Score и примерные веса

Компонент	Входные данные	Вес (примерный)
Freshness	Давность апдейта, сдвиг SERP, LinkVelocity (из GSC [17])	30%
Familiarity / Semantic Distance [14]	Покрытие СЯ, Е-Е-А-Т сигналы [14], семантическое расстояние [18] от ТОП-10 (из Semantic Service)	40%
Quality	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/ 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 with 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 внедрено, когда, кем/как (auto или HITL).

4.1.6. Требования к интеграции: адаптеры и плагины

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Для внедрения изменений на стороне клиента должны быть предоставлены следующие адаптеры/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

Tilda Adapter:

Реализация интеграции с Tilda через Webhook и API для автоматизированного обновления контента и мета-данных в рамках HITL-процесса.

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 HITL-одобрения.
- **Внешние API:** Данные, получаемые через API (GSC, GA4, PSI, LLM-провайдеры).
- **Сырые Данные:** HTML-код, CSS, JS, данные о ссылках, собранные собственным краулером.

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 HITL-одобрений с DiffViewer.
- **Отчеты:** CSV-экспорт сырых данных и аналитических агрегатов.
- **Изменения на Сайте Клиента:** Безопасные PATCH-операции, внедренные через Client API Gateway (WordPress, Tilda).

4.1.9. Требования к отчётности

- Оперативный отчет (еженедельный): позиции по кластерам, CTR, CWV (доля Good), ошибки сканирования, статус HITL-одобрений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 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 CSV (ориентир по модулям Reporting Service):

- 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 (WordPress, Tilda)

4.1.11. Требования к временным характеристикам

1. Краулинг и обработка:

- Полный цикл для 100 000 страниц: 20-30 часов (при 5 параллельных Playwright воркерах)
- Приоритизация: сначала страницы с органическим трафиком (данные из GSC)
- Incremental crawling: при повторных запусках обрабатываются только измененные страницы (проверка через Last-Modified header или ETag)

2. FF-Score пересчет:

- Ежедневный пересчет для всех активных проектов: не более 1 часа
- Real-time пересчет при HITLApproved: не более 30 секунд

3. API Gateway:

- Обработка HITL-запроса (подтверждение/отклонение): < 500 мс (включая запись в БД)
- Пропускная способность: 100 одновременных запросов

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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

(например, 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 «Сторона-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.

4.1.12.1. Публичный интерфейс первичного аудита

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Компоненты:

1. Лендинговая страница (/)

- Форма ввода URL сайта
- Краткое описание возможностей сервиса
- Примеры проверок

2. Страница результатов быстрого аудита (/audit/[uid])

- Визуализация базовых проблем (иконки, индикаторы)
- Список найденных ошибок (категории: критические, важные, рекомендации)
- СТА-блок: «Получить полный аудит → Регистрация»
- Возможность поделиться ссылкой на результат

3. Технические требования:

- Адаптивность: 320px — 1920px
- Время загрузки: LCP < 2.5 сек
- SSR для SEO-оптимизации лендинга
- Отдельный роутинг для публичных страниц (без авторизации)

Интеграция:

- API-эндпоинт: POST /api/public/quick-audit
- Polling результатов: GET /api/public/audit-status/[uid]

4.2. Требования к техническому обеспечению

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 данных краулинга).

4.3. Требования к маркировке и упаковке

Программный продукт поставляется в виде электронного комплекта, включающего программную документацию, само приложение (исполняемые 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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

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
Операционная Система	Linux (Ubuntu/Debian)
Backend	Python 3.10+, FastAPI
База Данных	PostgreSQL 14+ (с поддержкой JSONB)
Очередь Задач	Celery, Redis/RabbitMQ
Контейнеризация	Docker, Docker Compose
Краулинг	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.

4.6.1. Детализация HMAC-аутентификации

1. **Генерация подписи:** signature = HMAC-SHA256(secret_key, timestamp + method + url + body_hash)
2. **Заголовки запроса:** X-Signature: X-Timestamp: X-Project-ID:
3. **Валидация на сервере:**
 - Проверка временного окна: timestamp не старше 5 минут
 - Сравнение подписи через secure_compare (защита от timing attacks)
 - Проверка project_id в whitelist
4. **Ротация ключей:**
 - Автоматическая ротация каждые 90 дней
 - Grace period 7 дней: оба ключа (старый + новый) валидны
 - Уведомление администратора за 14 дней до ротации (email alert)
5. **Хранение ключей:**

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- Production: HashiCorp Vault или AWS Secrets Manager
- Development: .env файлы с правами 600 (не коммитятся в Git)

4.7. Требования к отказоустойчивости и мониторингу

1. High Availability:

- Frontend SPA: деплой на CDN (Cloudflare/Vercel) с 99.9% uptime SLA
- Backend микросервисы: минимум 2 реплики каждого сервиса (Docker Swarm/Kubernetes)
- PostgreSQL: streaming replication (primary + 1 replica)

2. Health Checks:

- Каждый микросервис предоставляет /health эндпоинт (возвращает JSON с версией, uptime)
- Проверка каждые 30 секунд; при 3 неудачных проверках — автоматический перезапуск

3. Мониторинг (Prometheus + Grafana):

Обязательные метрики:

- request_duration_seconds (histogram) — latency API запросов
- event_processing_lag_seconds — задержка обработки событий в очередях
- llm_api_errors_total (counter) — количество ошибок LLM API
- crawl_pages_processed_total (counter) — прогресс краулинга

Алерты:

- Error rate > 5% за 5 минут → Slack notification
- Queue lag > 10 минут → Email + PagerDuty
- Database connections > 80% pool size → Auto-scaling

4. Backup и Recovery:

- PostgreSQL: automated backups каждые 6 часов (retention 7 дней)
- Recovery Time Objective (RTO): < 4 часа
- Recovery Point Objective (RPO): < 6 часов (допустимая потеря данных)

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:** Покрывание кода Unit- и Integration-тестами.

4.9. Специальные требования

Особые требования к программе отсутствуют.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 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.17701729.05.13-02.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. Снижение зависимости от внешних API-сервисов (собственный краулер);
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 теми решениями, которые реально выбирает владелец сайта на практике: отдельные платформы для данных и семантики, отдельные краулеры для техаудита и отдельные системы “продвижения под ключ”.

Чтобы сравнение было честным и полезным, конкуренты сгруппированы по трём сценариям использования: аналитика/семантика, техаудит, автоматизация продвижения.

Внутри каждого сценария использовался утверждённый список конкурентов, сравниваемых по 5–7 критериям, которые напрямую влияют на ценность для пользователя (скорость старта, понятность результата, возможность регулярной работы, интеграции и отчётность).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Оценка ведется по шкале «Да / Частично / —», где «—» означает, что критерий не оценивался в рамках данной таблицы (либо не является типовым для класса).

Источники для заполнения — публичные страницы продуктов и/или документация, а также описание состава модулей и архитектуры разрабатываемой системы (Audit/Semantic/Reporting/Management, HITL-контур, API-шлюзы, адаптеры WordPress/Tilda). [25-48]

Таблица 5

SEO-платформы (данные/аналитика)

Критерий (данные/аналитика)	Ahrefs [25]	Semrush [26, 31]	MegaIndex [27, 32]	Rush Analytics [28]	Key Collector [29]	Наша раз- работка
Публичный API данных	Да	Да	Да	—	—	Да
Мониторинг позиций	Да	Да	Да	Да	—	Да
Данные по ссылкам (че- рез API/индекс)	Да	—	Да	—	—	—
Сбор/парсинг ключевых фраз	—	—	—	—	Да	Да

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 6

Технический аудит/краулинг

Критерий (техаудит)	Наша разработка (Audit Service)	Screaming Frog [33]	Netpeak Spider [34]
Краулинг JS/SPA (рендеринг)	Да	Да	Да
Интеграции с GA и GSC	Да	Да	Да
Автоматизация запусков	Да	Да	—
Экспорт/выгрузка результатов	Да	Да	Да
Параллельная работа с проектами	Да	—	Да

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 7

Автоматизация продвижения + “SEO из панели”

Критерий (автоматизация продвижения)	Наша разработка	PromoPult [35-38]	Rookee [39-42]	WebEffector [43]	REG.RU (авто-SEO/Seowizard) [44-46]	Timeweb (продвижение в панели) [47-48]
Запуск из ЛК/панели	Да	Да	Да	Да	Да	Да
Аудит сайта + рекомендации	Да	Частично	Да	Да	Да	Частично
Подбор/формирование запросов	Да	Да	—	Да	Да	—
Расчёт/настройка бюджета	Да	Да	Да	—	Да	—
Мониторинг позиций	Да	Да	Да	Да	Да	—
Прозрачная отчётность/аналитика	Да	Частично	Да	Да	Да	—

Сравнение показывает, что рынок в основном закрывает SEO “по частям”: одни продукты сильны в данных и мониторинге, другие — в краулинге и выявлении технических проблем, третьи — в запуске продвижения как услуги.

Разработка проектируется как сквозной контур: задачи выявляются (Audit/Semantic), приоритизируются (Management Service), и результат фиксируется в метриках/отчётности (Reporting), что сокращает время от “нашли проблему” до “влияние на KPI стало видно”.

Отдельный конкурентный класс — “SEO из панели хостинга/регистратора”: он выигрывает в простоте старта для непрофессионала, поэтому учитывается в сравнении как ближайшая альтернатива

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

по UX и порогу входа, а не как технический SEO-инструмент.
Ключевое отличие разработки — управляемость и безопасность изменений: HITL-подход и внедрение через шлюз/адаптеры позволяют масштабировать оптимизацию без потери контроля и без ручной рутины на стороне владельца сайта

6.3.1. Ограничения и риски разрабатываемой системы

Технические ограничения:

- 1. Масштабируемость краулера:
 - Playwright требует 500 МБ RAM на инстанс
 - При > 50 параллельных воркерах необходима кластеризация (Kubernetes)
- 2. Зависимость от внешних API:
 - LLM API (OpenAI/Gemini): при недоступности система работает в режиме degraded
 - GSC API: ограничение 1200 запросов/мин (проблема для > 100 проектов)
- 3. Точность авторских метрик:
 - FF-Score и E-E-A-T Score — эвристики, требующие валидации
 - Рекомендуется A/B тестирование на контрольной группе сайтов (6-12 месяцев)

Организационные риски:

- 1. Maintenance burden:
 - Обновление зависимостей (Playwright, Scrapy) каждые 3 месяца
 - Мониторинг изменений алгоритмов Google (Core Updates)
- 2. Compliance:
 - При коммерческом использовании: необходимость юридической проверки GDPR/ССРА
 - Риск блокировки IP при агрессивном краулинге чужих сайтов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 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.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 8

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
		Разработка и согласование технического задания с научным руководителем	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 – 10.01.26
		Формализация логики HITL-механизма	Зобов Арсений Александрович	17.12.25 – 10.01.26
	Разработка программы	Реализация Management Service и API Gateway	Зобов Арсений Александрович	11.01.26 – 25.02.26
		Реализация адаптеров для WordPress и Tilda	Зобов Арсений Александрович	01.03.26 – 20.03.26

Продолжение табл. 8

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.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 – 07.04.26
		Корректировка программы и программной документации по результатам испытаний	Зобов Арсений Александрович	18.04.26 – 26.04.26
Внедрение	Подготовка и передача программы	Подготовка программы и программной документации для презентации и защиты	Зобов Арсений Александрович	27.04.26 – 17.05.26
		Загрузка материалов курсового проекта в ЛМС, дисциплина «Курсовой проект, 2 курс, ПИ»	Зобов Арсений Александрович	04.05.26
		Защита программного продукта комиссии	Зобов Арсений Александрович	18.05.26 - 23.05.26

,

7.2. Сроки разработки и исполнители

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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.05.13-02.ТЗ 03-1

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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.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.17701729.05.13-02.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.17701729.05.13-02.ТЗ 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. Documentation — Ahrefs API : documentation [Электронный ресурс]. — URL: <https://ahrefs.com/api/documentation> (дата обращения: 03.02.2026).
26. Semrush API Help [Электронный ресурс]. — URL: <https://www.semrush.com/kb/5-api> (дата обращения: 03.02.2026).
27. API — MegaIndex [Электронный ресурс]. — URL: <https://megaindex.com/api> (дата обращения: 03.02.2026).
28. Bulk Rank Checker Tool — Rush Analytics [Электронный ресурс]. — URL: <https://rush-analytics.com/land/bulk-rank-checker> (дата обращения: 03.02.2026).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

29. Сбор фраз — Key Collector (документация) [Электронный ресурс]. — URL: <https://www.key-collector.ru/docs/data-collection/keywords/> (дата обращения: 03.02.2026).
30. Rank Tracker by Ahrefs: Track Keyword Rankings Globally [Электронный ресурс]. — URL: <https://ahrefs.com/rank-tracker> (дата обращения: 03.02.2026).
31. Utilizing Semrush for Position Monitoring (Position Tracking) [Электронный ресурс]. — URL: <https://subscribed.fyi/blog/position-tracking-in-semrush-utilizing-semrush-for-position-monitoring/> (дата обращения: 03.02.2026).
32. Online Visibility — MegaIndex API (visrep) [Электронный ресурс]. — URL: <https://megaindex.com/api/catalog/view/7> (дата обращения: 03.02.2026).
33. Screaming Frog SEO Spider Website Crawler [Электронный ресурс]. — URL: <https://www.screamingfrog.co.uk/seo-spider/> (дата обращения: 03.02.2026).
34. Netpeak Spider — In-Depth SEO Audit with Website Spider Tool [Электронный ресурс]. — URL: <https://netpeaksoftware.com/spider> (дата обращения: 03.02.2026).
35. Что такое система PromoPult [Электронный ресурс]. — URL: <https://promopult.ru/promopult> (дата обращения: 03.02.2026).
36. Добавление продвигаемого сайта в систему PromoPult (справка) [Электронный ресурс]. — URL: <https://help.promopult.ru/dobavlenie-prodvigaemogo-sajta-v-sistemu/> (дата обращения: 03.02.2026).
37. PromoPult: инструменты автоматизации SEO и ... (обзор) [Электронный ресурс]. — URL: <https://tods-blog.com.ua/seo-smo/instrumenty-seopult/> (дата обращения: 03.02.2026).
38. Обзор + отзыв на PromoPult — автоматизация продвижения [Электронный ресурс]. — URL: <https://gruzdevv.ru/services/promopult/> (дата обращения: 03.02.2026).
39. SEO: поисковая оптимизация и продвижение сайтов — Rookee [Электронный ресурс]. — URL: <https://rookee.ru/vitrina-uslug/seo/> (дата обращения: 03.02.2026).
40. О сервисе Rookee [Электронный ресурс]. — URL: <https://rookee.ru/about-us> (дата обращения: 03.02.2026).
41. Как продвинуть свой сайт самостоятельно: сервисы ... — блог Rookee [Электронный ресурс]. — URL: <https://rookee.ru/blog/kak-prodvinut-sayt/> (дата обращения: 03.02.2026).
42. Rookee (официальный сайт) [Электронный ресурс]. — URL: <https://rookee.ru> (дата обращения: 03.02.2026).
43. WebEffector — самостоятельное продвижение сайтов (официальный сайт) [Электронный ресурс]. — URL: <https://www.webeffector.ru> (дата обращения: 03.02.2026).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

44. Автоматическое SEO-продвижение — База знаний REG.RU [Электронный ресурс]. — URL: <https://help.reg.ru/support/pochta-i-servisy/servisy/avtomaticheskoye-seo-prodvizheniye> (дата обращения: 03.02.2026).
45. Сервис автоматического SEO-продвижения от REG.RU (Seowizard) [Электронный ресурс]. — URL: <https://www.reg.ru/web-tools/advertising/seowizard> (дата обращения: 03.02.2026).
46. REG.RU предлагает своим клиентам автоматическое SEO-продвижение (публикация) [Электронный ресурс]. — URL: <https://cmsmagazine.ru/journal/news-partners-1028/> (дата обращения: 03.02.2026).
47. SEO и контекстная реклама прямо в панели управления! — Timeweb (новость) [Электронный ресурс]. — URL: <https://timeweb.com/ru/about/news/2515/> (дата обращения: 03.02.2026).
48. Панель управления хостингом — документация Timeweb [Электронный ресурс]. — URL: <https://timeweb.com/ru/docs/virtualnyj-hosting/obshchaya-informaciya-o-hostinge/panel-upravleniya-hostingom/> (дата обращения: 03.02.2026).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.13-02.ТЗ 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.17701729.05.13-02.ТЗ 03-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]