Техническое задание

# 1. Наименование и область применения

**1.1. Наименование** Программное изделие: **"ContinueWork: Сервис автоматизации отбора кандидатов на основе обработки естественного языка"** .

**1.2. Краткая характеристика** Программное изделие представляет собой сервис, реализующий автоматизацию процесса отбора кандидатов на основе анализа текстовых данных из резюме и вакансий с использованием методов обработки естественного языка (NLP). Основными компонентами являются:

* Модуль векторизации текстов с применением модели **BERT** для анализа семантической близости;
* Система сопоставления резюме и вакансий на основе контекстного понимания ключевых навыков, опыта и требований;
* Интеграция с базами данных для хранения и обработки структурированных и векторных данных.

**1.3. Область применения** Программное изделие предназначено для автоматизации этапов первичного отбора кандидатов в процессе рекрутмента в **джоб-борды, отделы кадров, кадровые агентства** и другие организации, осуществляющие подбор персонала. Оно применяется в:

* **HR-системах и платформах управления персоналом** (например, порталах подачи резюме и вакансий);
* **Внутренних рекрутмент-системах** организаций;
* **Автоматизированных рабочих процессах** , требующих анализа текстовых данных для оценки соответствия кандидатов вакансиям.

Сервис реализует следующие функции:

1. **Анализ и ранжирование резюме** на основе семантического анализа текстов с использованием методов обработки естественного языка (NLP), включая векторизацию моделями типа BERT.
2. **Ускорение фильтрации** больших объемов неструктурированных данных (резюме, вакансии) за счет автоматизации этапов извлечения ключевых навыков, опыта и требований.
3. **Снижение влияния человеческого фактора** благодаря точному сопоставлению семантических векторов текстов, что минимизирует субъективные ошибки при оценке соответствия кандидатов.

Программное обеспечение интегрируется в ИТ-инфраструктуру клиентов для оптимизации процессов сопоставления резюме и вакансий, обеспечивая технологическую поддержку в контексте цифровизации рекрутмент-процедур.

# **2. Основание для разработки**

**2.1. Документы, на основании которых ведётся разработка**

1. Научно-исследовательская работа (НИР) № М24-534:  
   **«Программная реализация сервиса для автоматизации отбора кандидатов на основе обработки естественного языка из резюме и вакансий»** .  
   Пояснительная записка к НИР, подготовленная студентом Носовым А.И. под руководством Киреева В.С.

**2.2. Организация, утвердившая документ, и дата утверждения**

* **Организация** :  
  ФГАОУ ВО «Национальный исследовательский ядерный университет «МИФИ»  
  (Институт интеллектуальных кибернетических систем, Кафедра кибернетики №22).
* **Дата утверждения** : 2024 г. (на основании данных, указанных в пояснительной записке).

**2.3. Наименование и условное обозначение темы разработки**

* **Наименование темы** :  
  «Программная реализация сервиса для автоматизации отбора кандидатов на основе обработки естественного языка из резюме и вакансий».
* **Условное обозначение темы** : НИР М24-534.

# 3. Назначение разработки

#### **3.1 Функциональное назначение**

Программное изделие предназначено для автоматизации процесса первичного отбора кандидатов на основе анализа текстовых данных из резюме и вакансий с использованием методов обработки естественного языка (NLP). Основные функции включают:

1. **Анализ и векторизация текстов** :
   * Векторизация текстов резюме и вакансий с применением модели **BERT** для выявления семантической близости.
   * Извлечение ключевых навыков, опыта работы, образования и требований из неструктурированных текстов.
2. **Сопоставление резюме и вакансий** :
   * Ранжирование резюме по уровню соответствия требованиям вакансий на основе контекстного анализа.
   * Генерация списка кандидатов с оценкой степени соответствия (например, в виде процентной вероятности).
3. **Интеграция с базами данных** :
   * Хранение и обработка структурированных и векторных данных в реляционных базах (PostgreSQL, MongoDB) и векторных СУБД (Weaviate).
4. **Взаимодействие с внешними системами** :
   * API для интеграции с HR-платформами, порталами поиска работы и другими системами управления персоналом.
5. **Визуализация результатов** :
   * Предоставление рекрутерам отсортированных списков кандидатов с детализированными рекомендациями.

#### **3.2 Эксплуатационное назначение**

Программное изделие эксплуатируется в условиях:

1. **Среда развертывания** :
   * Облачные или локальные серверы с поддержкой контейнеризации (Docker).
   * Совместимость с операционными системами: Linux, Windows, macOS.
2. **Требования к аппаратному обеспечению** :
   * Процессор: минимум 4 ядра.
   * Оперативная память: не менее 8 ГБ.
   * Видеокарта с поддержкой GPU (для ускорения обработки моделей BERT).
3. **Требования к программному обеспечению** :
   * Пакеты Python (версии 3.8+), библиотеки для NLP (BERT, TensorFlow/PyTorch).
   * СУБД: PostgreSQL (для структурированных данных), Weaviate (для векторных данных).
4. **Условия использования** :
   * Обработка текстов на русском и английском языках.
   * Поддержка форматов: PDF, DOCX, TXT для загрузки резюме и вакансий.
   * Масштабируемость для работы с большими объемами данных (от 1000 до 10 000 записей в час).
5. **Безопасность** :
   * Шифрование данных (SSL/TLS для передачи, AES-256 для хранения).
   * Соответствие требованиям GDPR по обработке персональных данных.

# 4. Технические требования к программе или программному изделию

## **4.1. Состав выполняемых функций** Программное изделие должно обеспечивать выполнение следующих функций:

1. **Обработка текстовых данных резюме и вакансий:**
   * Векторизация текстов резюме и вакансий с использованием модели **BERT** для анализа семантической близости.
   * Извлечение ключевых навыков, опыта работы, образования и требований из неструктурированных текстов.
   * Поддержка обработки текстов на русском и английском языках.
2. **Сопоставление резюме и вакансий:**
   * Ранжирование резюме по уровню соответствия требованиям вакансий на основе контекстного анализа.
   * Генерация списка кандидатов с оценкой соответствия (например, в виде процентной вероятности).
   * Построение векторных представлений для резюме и вакансий с использованием методов NLP.
3. **Интеграция с базами данных:**
   * Хранение и обработка структурированных данных в реляционных СУБД (PostgreSQL).
   * Хранение векторных представлений в векторных СУБД (Weaviate).
   * Поддержка форматов данных: JSON для метаданных, векторные массивы для embeddings.
4. **Взаимодействие с внешними системами:**
   * API для интеграции с HR-платформами, порталами поиска работы и системами управления персоналом.
   * Поддержка RESTful API для обмена данными.
5. **Визуализация результатов:**
   * Предоставление рекрутерам отсортированного списка кандидатов с детализированными рекомендациями.
   * Генерация отчетов о процессе отбора в форматах PDF и CSV.

**4.1.2. Организация входных и выходных данных**

* **Входные данные:**
  + Текстовые данные резюме и вакансий в форматах: PDF, DOCX, TXT.
  + Структурированные данные (метаданные: дата публикации, категория вакансии и т.д.).
  + Запросы от пользователей через API для анализа конкретных вакансий или резюме.
* **Выходные данные:**
  + Список кандидатов, отсортированный по уровню соответствия вакансии.
  + Оценка соответствия (например, в виде числового рейтинга от 0 до 1).
  + Векторные представления текстов для хранения в базе данных.
  + Отчеты в форматах PDF и CSV для анализа результатов.

**4.1.3. Временные характеристики**

1. **Время обработки данных:**
   * Обработка 1 000 записей (резюме + вакансии) — не более 5 минут.
   * Ответ на запрос API (анализ одной пары «резюме-вакансия») — не более 2 секунд.
2. **Прогрузка данных:**
   * Импорт 10 000 записей в базу данных — не более 10 минут.
3. **Масштабируемость:**
   * Поддержка обработки до 10 000 записей в час при одновременной работе 100 пользователей.

**4.1.4. Дополнительные требования**

* **Интеграция с системами:**
  + Поддержка интеграции с популярными HR-системами (например, HeadHunter, LinkedIn Talent Solutions).
* **Обновление моделей:**
  + Возможность переключения между версиями моделей BERT и их переподгона под конкретные данные.
* **Логирование:**
  + Запись действий пользователей и ошибок в лог-файлы для анализа и отчетности.

## 4.2. Требования к надёжности

**4.2.1. Обеспечение устойчивого функционирования**

1. Программное изделие должно обеспечивать бесперебойную работу при обработке данных объемом до 10 000 записей в час.
2. Система должна поддерживать работу в распределенной архитектуре с изоляцией компонентов для минимизации влияния отказов отдельных сервисов.
3. Реализация механизма резервного копирования и восстановления данных:
   * Резервные копии должны создаваться ежедневно и храниться в зашифрованном виде.
   * Восстановление данных должно быть выполнено в течение **не более 15 минут** после выявления сбоя.
4. Мониторинг состояния системы в реальном времени с автоматическим оповещением администраторов при возникновении критических ошибок.

**4.2.2. Контроль входной и выходной информации**

1. **Входные данные** :
   * Валидация форматов файлов (PDF, DOCX, TXT) перед загрузкой.
   * Проверка целостности данных (например, через контрольные суммы).
   * Фильтрация SQL-инъекций и XSS-атак при обработке пользовательских вводов.
2. **Выходные данные** :
   * Валидация корректности формата отчетов (PDF, CSV).
   * Проверка корректности векторных представлений перед сохранением в БД.
3. **Контроль ошибок** :
   * Автоматический логгирование ошибок обработки с указанием причины, времени и контекста.

**4.2.3. Время восстановления после отказа (MTTR)**

* Время восстановления системы после критического сбоя должно составлять **не более 15 минут** .
* Для этого реализуются:
  + Автоматизированные процессы восстановления через облачные инфраструктуры (например, Kubernetes).
  + Репликация данных между географически распределенными узлами.

**4.2.4. Безопасность и соответствие стандартам**

1. **Шифрование данных** :
   * Данные в покое шифруются алгоритмом AES-256.
   * Данные в передаче защищаются протоколом TLS 1.3.
2. **Соблюдение GDPR** :
   * Обработка персональных данных должна соответствовать требованиям регламента GDPR.
   * Реализация механизма анонимизации данных при необходимости.
3. **Защита от несанкционированного доступа** :
   * Ролевая модель доступа (RBAC) для ограничения прав пользователей.
   * Многофакторная аутентификация для административных функций.

**4.2.5. Тестирование и мониторинг**

1. **Нагрузочное тестирование** :
   * Система должна поддерживать обработку до 10 000 записей в час без падения производительности.
   * Тестирование на устойчивость к одновременным запросам от 100 пользователей.
2. **Мониторинг** :
   * Сбор метрик производительности (CPU, память, время обработки запросов).
   * Интеграция с системами вроде **ELK Stack** или **Prometheus** для анализа логов и трендов.

## 4.3. Условия эксплуатации

**4.3.1. Физические условия эксплуатации** Программное изделие должно обеспечивать заданные характеристики при следующих условиях:

1. **Температура окружающего воздуха** :
   * В диапазоне от **+15°C до +30°C** для серверного оборудования.
   * Для рабочих станций пользователей: **от +10°C до +35°C** .
2. **Относительная влажность воздуха** :
   * **20% до 80%** (без конденсации).
3. **Электрические параметры** :
   * Напряжение питания: **110–240 В** (стабилизированное).
   * Частота тока: **50/60 Гц** .
4. **Климатические условия** :
   * Отсутствие агрессивных химических веществ, пыли, вибраций и ударных нагрузок.
   * Для облачных решений — соответствие стандартам дата-центров (ISO/IEC 27001).

**4.3.2. Требования к аппаратному обеспечению** Система должна эксплуатироваться на оборудовании, соответствующем следующим требованиям:

1. **Серверное оборудование** :
   * Процессор: не менее **4 ядер** (рекомендуется 8 и более для GPU-ускорения).
   * Оперативная память: **минимум 8 ГБ** (рекомендуется 16 ГБ и более для работы с большими объемами данных).
   * Видеокарта: поддержка **GPU** (например, NVIDIA Tesla или AMD Radeon Pro) для ускорения обработки моделей BERT.
   * Хранилище: SSD-диски с емкостью **не менее 100 ГБ** (в зависимости от объема данных).
2. **Клиентские устройства** :
   * Операционные системы: **Linux, Windows (10/11), macOS (10.15 и выше)** .
   * Браузеры: Chrome 90+, Firefox 88+, Edge 90+.
   * Для загрузки/выгрузки данных: поддержка форматов **PDF, DOCX, TXT** .

**4.3.3. Требования к сетевым условиям**

1. **Доступ к данным** :
   * Стабильное подключение к сети для работы с облачными сервисами (например, Weaviate, PostgreSQL).
   * Скорость интернета: **минимум 10 Мбит/с** (рекомендуется 100 Мбит/с и выше).
2. **Безопасность** :
   * Защита каналов связи протоколом **TLS 1.3** .
   * Ролевая модель доступа (RBAC) для ограничения прав пользователей.

**4.3.4. Вид обслуживания**

1. **Техническое обслуживание** :
   * Регулярное обновление ПО (включая ядра, драйверы GPU, библиотеки Python).
   * Мониторинг производительности и стабильности через инструменты вроде **Prometheus** или **ELK Stack** .
   * Резервное копирование данных ежедневно, хранение копий в зашифрованном виде.
2. **Обслуживание данных** :
   * Регулярная очистка баз данных от устаревших записей.
   * Оптимизация запросов и индексов в PostgreSQL и Weaviate.

**4.3.5. Квалификация и количество персонала** Для эксплуатации программного изделия требуется:

1. **Системный администратор (1–2 человека)** :
   * Опыт работы с Docker, Kubernetes, облачными платформами (AWS, Azure, Yandex Cloud).
   * Знание настройки и мониторинга серверов.
2. **Специалист по данным (1 человек)** :
   * Опыт администрирования PostgreSQL и векторных СУБД (Weaviate).
   * Навыки работы с большими данными и их анализа.
3. **Разработчик (1 человек)** :
   * Знание Python, фреймворков для NLP (BERT, TensorFlow/PyTorch).
   * Умение настраивать и переподготавливать модели BERT.

## 4.4. Требования к информационной и программной совместимости

##### **4.4.1. Требования к информационным структурам**

1. **Входные данные**
   * **Форматы файлов** :
     + Резюме и вакансии принимаются в форматах **PDF, DOCX, TXT** .
     + Метаданные (структурированные данные) передаются в формате **JSON** .
   * **Структура данных** :
     + Текстовые данные резюме и вакансий должны содержать разделы:
       - Должность/название вакансии.
       - Опыт работы, навыки, образование.
       - Требования (для вакансий).
     + Метаданные включают идентификаторы, даты публикации, категории.
   * **Кодировка** :
     + Данные обрабатываются в кодировке **UTF-8** .
2. **Выходные данные**
   * **Форматы** :
     + Списки кандидатов — JSON или CSV с оценкой соответствия.
     + Векторные представления — массивы чисел в формате **NumPy** или **JSON** .
     + Отчёты — PDF (для печати) и CSV (для анализа).
   * **Структура выходных данных** :
     + Рейтинг кандидатов:  
        json
     + {  
        "candidate\_id": "string",  
        "vacancy\_id": "string",  
        "similarity\_score": "float",  
        "recommendation": "string"  
        }
     + Векторные данные: массивы размерностью **768** (BERT).

##### **4.4.2. Требования к программной совместимости**

1. **Языки программирования и версии** :
   * Основной язык: **Python 3.8+** .
   * Поддержка интерпретаторов: CPython 3.8–3.11.
2. **Библиотеки и фреймворки** :
   * **Для обработки NLP** :
     + Hugging Face Transformers (версия 4.26+ для BERT).
     + spaCy (версия 3.5+ для предобработки).
   * **Для машинного обучения** :
     + TensorFlow 2.10+ или PyTorch 1.13+.
   * **Для взаимодействия с БД** :
     + SQLAlchemy 2.0+ (PostgreSQL).
     + Weaviate Client 3.22+ (векторные данные).
   * **Для API** :
     + FastAPI 0.95+ (RESTful API).
3. **Взаимодействие с внешними системами** :
   * **API-интерфейсы** :
     + Использование **RESTful API** с протоколом **HTTPS/TLS 1.3** .
     + Формат обмена данными — **JSON** .
   * **Базы данных** :
     + Реляционная СУБД: **PostgreSQL 14+** .
     + Векторная СУБД: **Weaviate 1.16+** .
     + Хранение неструктурированных данных: **MongoDB 6.0+** (опционально).
4. **Версии программного обеспечения** :
   * **Операционные системы** :
     + Поддержка **Linux (Ubuntu 20.04+/CentOS 8+)** , **Windows 10/11** , **macOS 10.15+** .
   * **Драйверы и зависимости** :
     + CUDA 11.8+ для GPU-ускорения BERT.
     + Docker 20.10+ и Docker Compose 2.17+ для контейнеризации.
5. **Контейнеризация и инфраструктура** :
   * **Контейнеры** :
     + Развертывание через Docker с образами, совместимыми с **Docker Engine 20.10+** .
   * **Оркестрация** :
     + Поддержка **Kubernetes 1.25+** для масштабирования.

##### **4.4.3. Требования к исходным кодам и методам решения**

1. **Язык программирования и методы реализации** :
   * **Язык** : Python для основной логики, SQL для запросов к БД.
   * **Методы обработки** :
     + Векторизация текстов с использованием модели **BERT** .
     + Семантическое сопоставление через косинусное расстояние.
   * **Архитектура** :
     + Микросервисная архитектура по стандарту **C4 Model** .
     + Использование **RESTful API** для взаимодействия между компонентами.
2. **Стандарты разработки** :
   * **Кодирование** :
     + Соблюдение стандартов **PEP8** для Python.
     + Комментарии и документация к функциям через **Sphinx** или **Google Style** .
   * **Тестирование** :
     + Unit-тесты с использованием **pytest** .
     + Интеграционные тесты через **Postman** для API.
3. **Совместимость исходного кода** :
   * **Версионность зависимостей** :
     + Указание версий библиотек в **requirements.txt** .
   * **Кросс-платформенность** :
     + Код должен компилироваться и запускаться на Linux, Windows, macOS.
4. **Интеграция с внешними сервисами** :
   * **API-совместимость** :
     + Поддержка интеграции с HR-платформами через **OAuth 2.0** и **JWT** .
   * **Базы данных** :
     + Синхронизация с PostgreSQL через **SQLAlchemy ORM** .
     + Векторные данные хранятся в Weaviate с использованием индексов **HNSW** .

##### **4.4.4. Требования к безопасности и стандартизации**

1. **Защита данных** :
   * Шифрование передачи данных — **TLS 1.3** .
   * Шифрование хранения — **AES-256** для метаданных.
2. **Совместимость с стандартами** :
   * Соответствие **GDPR** при обработке персональных данных.
   * Использование **OpenAPI 3.0** для документации API.

## 4.5. Требования к маркировке и упаковке

##### **4.5.1. Требования к маркировке программного изделия**

1. **Общая маркировка** :
   * Программное изделие должно быть идентифицировано следующими элементами:
     + **Название** : «ContinueWork: Сервис автоматизации отбора кандидатов».
     + **Версия** : Указание номера версии (например, v1.0.0) в установочных файлах, Docker-образах и документации.
     + **Уникальный идентификатор** : Номер НИР (НИР М24-534) и дата релиза.
     + **Разработчик** : ФГАОУ ВО «НИЯУ МИФИ», Кафедра кибернетики №22.
     + **Лицензионные условия** : Ссылка на лицензионное соглашение (например, MIT License или Proprietary).
2. **Маркировка компонентов** :
   * Все модули, базы данных (PostgreSQL, Weaviate) и API-интерфейсы должны содержать метаданные с указанием:
     + Названия компонента.
     + Версии (совместимость с основной версией сервиса).
     + Авторские права и условия использования.
3. **Маркировка обновлений** :
   * Патчи и обновления должны включать:
     + Номер версии и дату релиза.
     + Описание изменений и критичности обновления (например, «обязательное», «рекомендуемое»).

##### **4.5.2. Требования к упаковке**

1. **Цифровая упаковка** :
   * **Форматы распространения** :
     + Установочные файлы (например, .exe, .deb, .rpm).
     + Docker-образы для контейнеризации (опубликованные в Docker Hub или частном репозитории).
     + ZIP-архивы с исходным кодом и документацией.
   * **Содержание упаковки** :
     + Основные компоненты:
       - Исполняемые файлы сервиса.
       - Конфигурационные файлы (.yaml, .env).
       - Скрипты для настройки и запуска (например, docker-compose.yml).
     + Документация:
       - Инструкции по установке и настройке.
       - API-документация (генерируемая с помощью Swagger или FastAPI).
       - Лицензионное соглашение.
2. **Условия хранения и транспортировки** :
   * Для физических носителей (если применимо):
     + Упаковка в коробки с защитой от повреждений и влаги.
     + Метки с указанием содержимого, версии и даты выпуска.
3. **Безопасность упаковки** :
   * Цифровые файлы должны быть подписаны электронной подписью разработчика.
   * Обновления и патчи распространяются через защищенные каналы (HTTPS, SFTP).

##### **4.5.3. Требования к документации в упаковке**

1. **Документы, включенные в упаковку** :
   * **Инструкция по установке** : Пошаговое руководство для развертывания в локальной или облаковой среде.
   * **Документация API** : Описание эндпоинтов и параметров.
   * **Лицензионное соглашение** : Условия использования программного обеспечения.
   * **Руководство по настройке** : Настройка под конкретный стек технологий (PostgreSQL, Weaviate).
2. **Форматы документов** :
   * Документация предоставляется в форматах **PDF** , **Markdown** и **HTML** для удобства доступа.

##### **4.5.4. Требования к контейнеризации (Docker)**

1. **Метки Docker-образов** :
   * Имя образа: continuework/hr-service:<версия>.
   * Теги версий: latest, <major>.<minor>.<patch> (например, v1.0.2).
   * Описание образа в Docker Hub:
     + Краткое назначение сервиса.
     + Требования к среде выполнения (например, GPU поддержка).
2. **Содержимое Docker-образов** :
   * Включает все зависимости и конфигурации для автономного запуска.
   * Скрипты для инициализации баз данных и настройки окружения.

##### **4.5.5. Требования к облачной упаковке (если применимо)**

* **Развертывание в Kubernetes** :
  + Манифесты Helm или YAML-файлы для оркестрации.
  + Метки ресурсов (название, версия, пространство имен).

## 4.6. Требования к транспортированию и хранению

##### **4.6.1. Условия транспортирования**

1. **Цифровые носители** :
   * Транспортировка должна осуществляться через защищенные каналы связи:
     + **Протоколы** : HTTPS, SFTP, FTPS.
     + **Шифрование** : Все данные передаются в зашифрованном виде (AES-256 для файлов, TLS 1.3 для передачи).
   * Docker-образы и исходный код распространяются через защищенные репозитории (например, Docker Hub, GitHub/GitLab Enterprise).
   * Цифровые копии подписываются электронной подписью разработчика для предотвращения подделки.
2. **Физические носители** (при наличии):
   * Упаковка должна обеспечивать защиту от влаги, пыли и механических повреждений.
   * Температура при транспортировке: **от -5°C до +40°C** .
   * Относительная влажность: **не более 80%** (без конденсата).
   * Запрещается транспортировка в условиях сильных вибраций или ударов.

##### **4.6.2. Условия хранения**

1. **Цифровые носители** :
   * **Облачные хранилища** :
     + Данные хранятся в инфраструктурах, соответствующих стандартам **ISO/IEC 27001** (например, AWS, Azure, Yandex Cloud).
     + Резервные копии защищены шифрованием (AES-256) и реплицируются в географически распределенных центрах данных.
   * **Локальные хранилища** :
     + Файлы хранятся на SSD-дисках с RAID-конфигурацией для защиты от потери данных.
     + Системы хранения поддерживают шифрование данных в покое (AES-256).
2. **Физические носители** (например, USB-накопители, оптические диски):
   * Температура: **от +10°C до +30°C** .
   * Влажность: **не более 60%** (без конденсата).
   * Хранение в герметичных контейнерах с защитой от пыли и света.
   * Место хранения должно быть охраняемым и иметь контроль доступа (RFID-метки, пароли).
3. **Данные в базах** :
   * **PostgreSQL и Weaviate** :
     + Хранение в защищенных средах с репликацией данных.
     + Регулярные резервные копии (ежедневно, еженедельно, ежемесячно).
   * **Модели BERT** :
     + Хранятся в защищенных облачных хранилищах (например, Amazon S3) с ограничением доступа.

##### **4.6.3. Условия складирования**

1. **Физические носители** :
   * Складирование в вертикальных стойках для предотвращения деформации.
   * Метки носителей должны содержать:
     + Название ПО («ContinueWork: Сервис автоматизации отбора кандидатов»).
     + Версия, дата выпуска и уникальный идентификатор (НИР М24-534).
   * Разделение носителей по критериям важности (например, критические данные в отдельных ячейках).
2. **Цифровые носители** :
   * Docker-образы и исходный код хранятся в изолированных средах с ограничением доступа по принципу «минимальных прав» (RBAC).
   * Модели BERT и векторные данные размещаются в специализированных хранилищах (Weaviate, Pinecone) с настройкой шардирования и репликации.

##### **4.6.4. Сроки хранения**

1. **Основная версия программного изделия** :
   * Хранится не менее **5 лет** с момента выпуска.
   * Рекомендуется ежегодное обновление до актуальных версий ПО и зависимостей.
2. **Резервные копии** :
   * **Ежедневные резервы** : Хранятся **30 дней** .
   * **Еженедельные резервы** : Хранятся **12 месяцев** .
   * **Ежемесячные архивные копии** : Хранятся **не менее 5 лет** .
3. **Модели BERT и векторные данные** :
   * Основные версии моделей хранятся в облачных хранилищах с автоматической репликацией.
   * Срок хранения моделей: **не менее 3 лет** с момента последнего использования.

##### **4.6.5. Требования к безопасности хранения**

1. **Цифровые носители** :
   * Доступ регулируется через многофакторную аутентификацию (MFA).
   * Контроль версий и изменений через системы управления конфигурациями (например, Git).
2. **Физические носители** :
   * Доступ ограничен лицам с соответствующими правами (RBAC).
   * Хранение в охраняемых помещениях с системами видеонаблюдения и контроля доступа.
3. **Данные в базах** :
   * PostgreSQL и Weaviate настроены на автоматическое шифрование данных в покое.
   * Логирование и мониторинг доступа к данным через инструменты вроде **ELK Stack** .

# 5. Технико-экономические показатели

#### **5.1. Ориентировочная экономическая эффективность**

Реализация сервиса автоматизации отбора кандидатов ожидается снизить затраты на процесс рекрутмента за счет:

1. **Сокращения времени первичного отбора** :
   * Уменьшение времени, затрачиваемого HR-специалистами на анализ резюме, на **50–60%** (по результатам тестирования, 85% релевантных кандидатов выявляются автоматически).
   * Снижение затрат на зарплату HR-персонала за счет оптимизации рабочих процессов.
2. **Снижение издержек на подбор персонала** :
   * Уменьшение затрат на поиск и отбор кандидатов на **30–40%** за счет уменьшения числа ошибок и повторных подборов.
   * Снижение риска потери квалифицированных кандидатов из-за задержек в процессе отбора.
3. **Повышение качества отбора** :
   * Увеличение точности соответствия кандидатов вакансиям до **85%** (по данным тестирования на реальных данных).
   * Снижение количества ошибок отбора на **40–50%** , что уменьшает затраты на обучение и адаптацию неподходящих сотрудников.

#### **5.2. Предполагаемая годовая потребность**

Система должна обеспечивать обработку следующих объемов данных в год:

1. **Обработка вакансий** :
   * Обслуживание **10 000+ вакансий** в год (включая их анализ и сопоставление с резюме).
2. **Обработка резюме** :
   * Анализ **100 000+ резюме** в год.
3. **Масштабируемость** :
   * Обработка до **10 000 записей в час** (резюме + вакансии) в пиковые нагрузки.
   * Поддержка одновременной работы с **100+ пользователями** (HR-специалистами).

#### **5.3. Экономические преимущества по сравнению с аналогами**

Сервис демонстрирует следующие преимущества по сравнению с существующими решениями:



| **Точность отбора** | 85% (MCC 0.55) | 65–75% (базовые алгоритмы) | 70–80% (зависит от используемых моделей) |
| --- | --- | --- | --- |
| **Скорость обработки** | 10 000 записей/час | 5000–7000 записей/час | 8000–10 000 записей/час (зависит от инфраструктуры) |
| **Стоимость владения** | Низкая (открытые модели BERT, облачные решения) | Средняя (платные API, лицензии) | Высокая (платные подписки, проприетарные технологии) |
| **Интеграция с системами** | Полная (REST API, Docker) | Частичная (ограниченные API) | Требует дорогостоящих настроек |

#### **5.4. Ожидаемая экономия**

1. **Снижение затрат на подбор персонала** :
   * Уменьшение расходов на HR-персонал за счет автоматизации первичного отбора.
   * Экономия **от 20 до 30%** бюджета на рекрутмент за счет сокращения времени и ошибок.
2. **Увеличение производительности HR-отделов** :
   * Освобождение HR-специалистов для работы над стратегическими задачами (например, интервью, анализ soft skills).
3. **Снижение затрат на вычислительные ресурсы** :
   * Использование открытых моделей BERT и оптимизированных алгоритмов сокращает расходы на обучение проприетарных моделей (экономия **15–20%** от бюджета на ИТ-инфраструктуру).

#### **5.5. Сравнение с аналогами**

1. **По сравнению с отечественными системами** :
   * **Преимущества** :
     + Высокая точность (BERT vs. традиционные TF-IDF/BoW).
     + Поддержка контекстного анализа (семантическое сопоставление vs. ключевые слова).
     + Низкая стоимость владения (открытые технологии vs. платные лицензии).
2. **По сравнению с зарубежными решениями** :
   * **Преимущества** :
     + Локализация под русскоязычные резюме и вакансии (например, корректное понимание терминологии).
     + Гибкость настройки (модели BERT могут переподготавливаться под конкретные требования компании).
     + Меньшие затраты на интеграцию с российскими HR-платформами (например,.hh.ru, Superjob).

#### **5.6. Ожидаемые сроки окупаемости**

* **Срок окупаемости** : **12–18 месяцев** за счет:
  + Сокращения затрат на подбор персонала.
  + Снижения издержек на обучение неподходящих кандидатов.
  + Уменьшения времени простоя позиций из-за задержек в отборе.

#### **5.7. Обоснование показателей**

1. **Эффективность BERT** :
   * Модель демонстрирует точность **85%** на реальных данных (см. раздел 4.3 пояснительной записки), что выше, чем у решений на основе TF-IDF или Word2Vec.
2. **Масштабируемость** :
   * Архитектура на основе Docker и Kubernetes позволяет горизонтально масштабировать систему (см. раздел 3.2).
3. **Стоимость** :
   * Использование открытых библиотек (Hugging Face, Weaviate) и облачных решений сокращает затраты на 30–40% по сравнению с проприетарными аналогами.

# 6. Стадии и этапы разработки

#### **6.1. Стадии разработки**

Разработка программного изделия осуществляется в следующих стадиях:

1. **Подготовительная стадия**
   * Анализ требований и выбор технологий.
   * Разработка технического задания и согласование с заказчиком.
2. **Проектирование**
   * Разработка архитектуры и системных требований.
   * Выбор СУБД и инструментов разработки.
3. **Реализация**
   * Программная разработка модулей (векторизации, сопоставления, API).
   * Интеграция с базами данных и контейнеризация.
4. **Тестирование и отладка**
   * Функциональное, нагрузочное и интеграционное тестирование.
   * Оценка метрик точности (accuracy, MCC) и производительности.
5. **Внедрение и сдача в эксплуатацию**
   * Подготовка документации и обучение пользователей.
   * Согласование результатов с заказчиком.

#### **6.2. Этапы разработки и содержание работ**



| **1. Подготовка** | - Анализ требований и существующих решений. | - Техническое задание. |
| --- | --- | --- |
|  | - Выбор модели BERT для векторизации текстов. | - Отчет по анализу аналогов. |
|  | - Определение сроков и ресурсов. | - План разработки. |
| **2. Проектирование** | - Разработка архитектуры (микросервисы, использование C4 модели). | - Архитектурные диаграммы (UML, BPMN). |
|  | - Выбор СУБД (PostgreSQL для структурированных данных, Weaviate для векторных данных). | - Требования к СУБД. |
|  | - Разработка функциональных и нефункциональных требований. | - Системные требования (раздел 4 ТЗ). |
| **3. Реализация** | - Разработка модулей: обработки текстов, API, взаимодействия с БД. | - Код программного обеспечения. |
|  | - Контейнеризация (Docker) и развертывание в Kubernetes. | - Руководства по установке и настройке. |
| **4. Тестирование** | - Функциональное тестирование (проверка сопоставления резюме и вакансий). | - Отчёты по тестированию (unit-, интеграционные тесты). |
|  | - Нагрузочное тестирование (обработка 10 000 записей/час). | - Результаты тестирования производительности. |
|  | - Оценка метрик accuracy (75%) и MCC (0.55) на реальных данных. | - Доклад по результатам экспериментов с моделью BERT. |
| **5. Внедрение** | - Документирование системы. | - Пользовательская инструкция. |
|  | - Обучение персонала. | - Документация по API и интеграции. |
|  | - Сдача системы в эксплуатацию. | - Акт сдачи-приемки. |

#### **6.3. Сроки разработки**

Разработка планируется в следующие сроки (примерные):

1. **Подготовка** — 1 месяц.
2. **Проектирование** — 2 недели.
3. **Реализация** — 2 месяца.
4. **Тестирование** — 1 месяц.
5. **Внедрение** — 2 недели.  
   **Итоговый срок** : **4,5 месяца** с момента согласования ТЗ.

#### **6.4. Исполнители**

1. **Разработчик** :
   * Носов Артём Иванович (студент, группа М24-534).
   * Основные обязанности: разработка кода, тестирование, документирование.
2. **Руководитель** :
   * Киреев Василий Сергеевич.
   * Основные обязанности: согласование требований, проверка промежуточных результатов, утверждение документов.
3. **Специалисты кафедры кибернетики №22** :
   * Участие в проверке архитектуры и соответствия стандартам.
4. **Тестировщики** :
   * Проверка функциональности и производительности системы.

#### **6.5. Документы, разрабатываемые на каждом этапе**

1. **Подготовительный этап** :
   * Техническое задание (утверждено).
   * Отчет по анализу аналогов.
2. **Проектирование** :
   * Системные требования.
   * Архитектурные диаграммы (C4, UML).
   * Спецификация API.
3. **Реализация** :
   * Код программного обеспечения.
   * Руководство по развертыванию.
4. **Тестирование** :
   * Отчёты по функциональным и нагрузочным тестам.
   * Отчет о результатах экспериментов с моделью BERT.
5. **Внедрение** :
   * Инструкция по эксплуатации.
   * Акт сдачи-приемки.

#### **6.6. Обоснование этапов**

1. **Подготовка** обеспечивает выбор оптимальных технологий (BERT, микросервисная архитектура) и согласование требований.
2. **Проектирование** фокусируется на создании надежной архитектуры с использованием PostgreSQL и Weaviate для хранения данных.
3. **Реализация** включает разработку модулей с применением инкрементного подхода, что позволяет поэтапно улучшать систему.
4. **Тестирование** проверяет соответствие метрикам точности (accuracy 75%, MCC 0.55) и требований к производительности (обработка 10 000 записей/час).
5. **Внедрение** гарантирует готовность системы к эксплуатации с документацией и обучением пользователей.

#### **6.7. Особенности жизненного цикла**

* **Инкрементный подход** :
  + Реализация и тестирование функций поэтапно (например, сначала векторизация текстов, затем интеграция с БД).
* **Совместимость с Agile** :
  + Регулярные итерации с демонстрацией результатов руководителю и заказчику.
* **Тест-драйв** :
  + Проверка моделей BERT на реальных данных (тестовый датасет из 172 321 записи).

# 7. Порядок контроля и приемки

#### **7.1. Виды испытаний**

Программное изделие проходит следующие виды испытаний для подтверждения соответствия требованиям:

1. **Функциональное тестирование** :
   * Проверка корректности выполнения всех заявленных функций:
     + Векторизация текстов резюме и вакансий с использованием BERT.
     + Ранжирование кандидатов по уровню соответствия вакансиям.
     + Взаимодействие с базами данных (PostgreSQL, Weaviate).
     + Обработка входных данных (PDF, DOCX, TXT) и формирование выходных отчетов (CSV, PDF).
   * Критерии приемки:
     + Точность сопоставления резюме и вакансий не менее **75%** (по метрике accuracy).
     + Коэффициент корреляции Мэтьюса (MCC) не менее **0.55** .
2. **Модульное тестирование** :
   * Проверка отдельных компонентов системы:
     + Модуль векторизации текстов.
     + Модуль сопоставления резюме и вакансий.
     + API-интерфейсы и взаимодействие с БД.
   * Критерии приемки:
     + Отсутствие критических ошибок (критерий: **0 ошибок** в unit-тестах).
3. **Интеграционное тестирование** :
   * Проверка взаимодействия компонентов системы:
     + Обмен данными между микросервисами (векторизация, сопоставление, хранение).
     + Интеграция с внешними системами (например, HR-платформами через REST API).
   * Критерии приемки:
     + Скорость обработки одной пары «резюме-вакансия» не более **2 секунд** .
     + Устойчивость системы при одновременном обработке **100+ запросов** .
4. **Нагрузочное тестирование** :
   * Проверка производительности при обработке больших объемов данных:
     + Обработка **10 000 записей** в час.
     + Импорт **10 000 записей** в базу данных за **не более 10 минут** .
   * Критерии приемки:
     + Система должна обрабатывать пиковые нагрузки без падения производительности.
5. **Тестирование безопасности** :
   * Проверка соответствия требованиям безопасности:
     + Шифрование данных в покое (AES-256) и в передаче (TLS 1.3).
     + Защита от SQL-инъекций, XSS-атак и несанкционированного доступа.
     + Соответствие GDPR для обработки персональных данных.
   * Критерии приемки:
     + Отсутствие уязвимостей, выявленных инструментами анализа (например, OWASP ZAP).
6. **Тестирование совместимости** :
   * Проверка работоспособности в следующих условиях:
     + Операционные системы: **Linux, Windows, macOS** .
     + Браузеры: Chrome 90+, Firefox 88+, Edge 90+.
     + Поддержка форматов: PDF, DOCX, TXT.
   * Критерии приемки:
     + Полная совместимость с указанными платформами и форматами.

#### **7.2. Общие требования к приемке работы**

1. **Требования к функциональности** :
   * Программное изделие должно полностью реализовать требования раздела **4.1** ТЗ.
   * Достижение метрик точности (accuracy 75%, MCC 0.55) на реальных данных (тестовый датасет из **172 321 записи** ).
2. **Требования к производительности** :
   * Обработка **1 000 записей** (резюме + вакансии) за **не более 5 минут** .
   * Ответ на запрос API — **не более 2 секунд** .
3. **Требования к безопасности** :
   * Соответствие стандартам **ISO/IEC 27001** для хранения и передачи данных.
   * Защита от утечек персональных данных согласно **GDPR** .
4. **Требования к документации** :
   * Присутствие следующих документов:
     + Инструкция по установке и настройке.
     + API-документация (сгенерированная с помощью Swagger/FastAPI).
     + Руководство по эксплуатации.
     + Отчеты по тестированию (функциональные, нагрузочные, безопасность).
   * Документация должна соответствовать **PEP8** и быть локализована на русском/английском языках.
5. **Требования к релизу** :
   * Успешное прохождение всех видов испытаний (см. раздел 7.1).
   * Устранение всех критических и серьезных дефектов до приемки.
   * Формирование **акта сдачи-приемки** с подписями разработчика и заказчика.

#### **7.3. Порядок приемки**

1. **Этап 1. Промежуточная приемка** :
   * Проводится после завершения этапа **реализации** (см. раздел 6.2 ТЗ).
   * Проверяются:
     + Корректность работы основных модулей (векторизация, сопоставление).
     + Документация по API и интеграции.
2. **Этап 2. Финальная приемка** :
   * Проводится после завершения этапа **тестирования** (см. раздел 6.2 ТЗ).
   * Проверяются:
     + Соответствие всем видам испытаний (функциональное, нагрузочное, безопасность).
     + Достижение требований к производительности и надежности (разделы 4.1, 4.2).
     + Готовность документации и установочных пакетов.
3. **Этап 3. Сдача в эксплуатацию** :
   * После успешного прохождения этапа 2, система передается заказчику.
   * Формируется **акт сдачи-приемки** с указанием:
     + Версии программного изделия.
     + Даты релиза и списка исправленных ошибок.
     + Подписями руководителя проекта и заказчика.

#### **7.4. Критерии приемки**

Система признается принявшейся, если:

1. Все функции работают в соответствии с ТЗ и требованиями раздела 4.
2. Производительность и надежность соответствуют разделам **4.1.3** и **4.2.3** (например, время восстановления после отказа — **не более 15 минут** ).
3. Безопасность данных подтверждена отчетами по тестированию и аудиту.
4. Документация полная и соответствует стандартам.

#### **7.5. Ответственные за приемку**

* **Руководитель проекта** : Киреев Василий Сергеевич (утверждение результатов тестирования).
* **Заказчик** : Представитель кафедры кибернетики №22 НИЯУ МИФИ.
* **Тестировщики** : Специалисты по автоматизации и безопасности.

#### **7.6. Обоснование**

Требования к контролю и приемке основаны на:

* Результатах тестирования (раздел 4.3 пояснительной записки), где показано достижение accuracy 75% и MCC 0.55.
* Системных требованиях к производительности (раздел 4.1.3) и безопасности (раздел 4.2).
* Требованиях к документации из раздела **6.5** (примеры: отчеты по тестированию, акт сдачи-приемки).

Этот раздел соответствует этапам разработки (раздел 6.2) и жизненному циклу с инкрементным подходом (раздел 4.1 пояснительной записки).