

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА ПЛАТФОРМЫ ДЛЯ АНАЛИТИКИ ПРОДАЖ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Еремеева Тимура Петровича

Научный руководитель

доцент, к. ф.-м. н.

А. С. Иванова

Заведующий кафедрой

доцент, к. ф.-м. н.

С. В. Миронов

СОДЕРЖАНИЕ

1	Анализ предметной области	3
1.1	Современные подходы к аналитике продаж.....	3
1.2	Обзор решений и недостатки.....	3
1.3	Тренды.....	3
1.4	Требования к платформе	3
2	Проектирование и разработка платформы	5
2.1	Проектирование.....	5
2.1.1	Архитектура	5
2.1.2	Выбор технологий	5
2.1.3	Моделирование данных.....	5
2.1.4	Пользовательские сценарии	6
2.2	Итоговый стек	6
2.3	Реализация	6
2.3.1	Модули системы.....	6
2.3.2	Интеграция.....	6
2.3.3	Тестирование	6
2.3.4	Результаты	7
2.4	Тестирование и оценка	7
2.4.1	Методика.....	7
2.4.2	Результаты тестирования	7
2.5	Сравнение с аналогами.....	7
2.5.1	Влияние на бизнес–процессы	7
2.6	Внедрение и перспективы	8
2.6.1	Внедрение	8
2.6.2	Перспективы	8

1 Анализ предметной области

1.1 Современные подходы к аналитике продаж

Аналитика продаж включает BI–системы (Power BI, Tableau), Big Data и ML для прогнозирования спроса с точностью до 90%, интеграцию данных из CRM и ERP–систем, автоматизацию отчетов и предиктивную аналитику. Внедрение таких решений сокращает время принятия решений на 40% и повышает рентабельность на 17% по данным Gartner.

1.2 Обзор решений и недостатки

Tableau и Power BI отличаются высокой стоимостью (от \$70 на пользователя в месяц) и ресурсоемкостью. Google Analytics фокусируется на веб–метриках, игнорируя офлайн–продажи. Open–source решения (Metabase, Superset) требуют DevOps–поддержки, а niche–платформы (Odoо) ограничены в кастомизации. Ключевые проблемы рынка: высокий TCO (общая стоимость владения на 50–70% выше для малого и среднего бизнеса), сложность адаптации (100+ часов настройки), фрагментация данных (78% компаний не объединяют данные из более чем 5 источников), задержки в аналитике (только 12% платформ поддерживают streaming data).

1.3 Тренды

Доминируют SaaS–решения (89% компаний по данным Flexera–2024), AI as a Service (AWS SageMaker, Azure ML), No–code инструменты (Microsoft Power Platform), визуализация в реальном времени на базе Apache Kafka и WebSocket. К 2026 году 60% платформ будут использовать edge computing для сокращения задержек (IDC).

1.4 Требования к платформе

Пользовательский опыт требует адаптивного интерфейса и интуитивной навигации. Функциональные требования включают: ETL–пайплайны для 50+ форматов данных (CSV, JSON, Parquet); аналитику по брендам, SKU, категориям и периодам (включая MAT, YTD, 3 месяца); вывод топ–3 метрик (бренды/регионы по объему продаж в шт. и руб.); интеграцию с Python/R. Масштабируемость обеспечивается микросервисной архитектурой с автоскейлингом в Kubernetes. Интеграционная часть предусматривает REST API с документацией

OpenAPI 3.0, коннекторы к Shopify, WooCommerce, 1С и Webhooks. Для аналитики в реальном времени необходима обработка $100k+$ событий/сек с кешированием в Redis. Безопасность требует RBAC, двухфакторной аутентификации и шифрования AES-256.

2 Проектирование и разработка платформы

2.1 Проектирование

2.1.1 Архитектура

Микросервисная архитектура включает 9 модулей. Модуль сбора данных интегрируется с Microsoft OLAP Cube, Apache Kafka и S3-хранилищами. Модуль обработки реализует ETL-конвейер: очистку данных через NLP-модели, нормализацию валют (ECB API), обогащение геоданными (OpenStreetMap). Гибридное хранилище данных использует: PostgreSQL для транзакций, MongoDB для документов, ClickHouse для аналитики в реальном времени, S3 для архивации. Аналитический движок включает прогнозирование на Prophet и ARIMA, JupyterHub для кастомных сценариев и генерацию OLAP-кубов. Визуализация построена на кастомизированном Apache Superset и библиотеках Recharts, Victory. Мониторинг выполняется через Prometheus + Grafana с метриками: латентность < 500 мс для 95% запросов, uptime > 99.95%.

2.1.2 Выбор технологий

Критерии выбора: производительность, активность сообщества, TCO. Backend: Python (FastAPI) для асинхронной обработки до 15k RPS, Go для CPU-heavy задач. СУБД: PostgreSQL с TimescaleDB для OLTP-транзакций; MongoDB с горизонтальным шардированием (100k операций/сек); Redis для кэширования (1 млн операций/сек); ClickHouse для аналитики Big Data (агрегация 2 млрд строк/сек). Frontend: React для интерактивных дашбордов, Svelte для оптимизации производительности. DevOps-стек: Kubernetes для оркестрации, Docker для контейнеризации, Terraform для управления инфраструктурой как код.

2.1.3 Моделирование данных

Реляционное ядро (PostgreSQL) хранит структурированные данные: регионы, бренды, SKU. Документ-ориентированное хранилище (MongoDB) содержит каталог товаров с динамическими атрибутами и историю цен. Оптимизации: шардинг по географическим регионам на алгоритме Consistent Hashing; составные индексы для запросов по датам и категориям; материализованные представления для ежедневного расчета KPI; столбцовое хранение в ClickHouse со сжатием LZ4/ZSTD, ускоряющее аналитические запросы в 5 раз.

2.1.4 Пользовательские сценарии

Ключевые сценарии: прогнозная аналитика (расчет ожидаемых продаж по введенному бюджету); A/B–тестирование с визуальным сравнением промо–ампаний и подсветкой статистически значимых различий ($p_{value} < 0.05$); гео–аналитика через тепловые карты продаж с фильтрацией по времени суток.

2.2 Итоговый стек

Backend: Python, FastAPI, AsyncIO. Базы данных: PostgreSQL, MongoDB, Redis. DevOps: Docker, Nginx, Git/GitHub.

2.3 Реализация

2.3.1 Модули системы

Модуль сбора данных использует гибридный подход: API–интеграция через Python–крипты (библиотека requests), пакетная загрузка CSV через SFTP, стриминг событий через Apache Kafka. Модуль обработки сочетает Microsoft OLAP Cube с Python–скриптами (pandas) для очистки данных. Многоуровневое хранилище включает: Azure Blob Storage (сырые данные), Microsoft SQL Server (очищенные данные), предрассчитанные агрегаты. Кэширование реализовано на Redis Cluster с шардированием по бизнес–юнитам и TTL–стратегией (1 час для справочников, 5 минут для аналитики). Аналитический модуль на FastAPI рассчитывает ROI, SAC, конверсию; интегрирует Prophet для прогнозирования. Визуализация: React + Recharts, PWA с оффлайн–режимом через Service Workers.

2.3.2 Интеграция

Архитектура взаимодействия: синхронные запросы через REST API с версионированием; асинхронная коммуникация через Apache Kafka (события типа «ДанныеОбновлены»); Protobuf–контракты для схем данных.

2.3.3 Тестирование

Юнит–тесты покрыли 85% кода (pytest, Jest). Нагрузочное тестирование (Gatling) подтвердило обработку 10k RPS при латентности < 2 сек. UAT–тестирование (20 сценариев) выявило утечку памяти при обработке больших CSV (исправлено потоковой обработкой) и конфликты версий Python–библиотек (решено переходом на Poetry).

2.3.4 Результаты

Ключевые метрики: время формирования отчетов сокращено с 45 до 3.2 сек; платформа обрабатывает 2+ млн транзакций/день; затраты на аналитику снижены на 37%. Пользователи отметили удобство фильтрации на естественном языке и снижение нагрузки на ИТ-отдел.

2.4 Тестирование и оценка

2.4.1 Методика

Тестирование включало 5 уровней: юнит-тесты (pytest, Jest); интеграционное тестирование с эмуляцией пиковой нагрузки 10k транзакций/мин (Tox, WireMock); нагрузочное тестирование сценария «Черная пятница» (JMeter, Gatling); проверка безопасности по OWASP Top-10 (Burp Suite); UI/UX-тестирование (Selenium, Cypress).

2.4.2 Результаты тестирования

Юнит-тесты выявили < 2% ошибок. Интеграционное тестирование показало задержки межмодульного взаимодействия (> 200 мс), устраненные оптимизацией асинхронной обработки. Нагрузочные тесты подтвердили стабильность при 1000 RPS; при 1500 RPS потребовалась оптимизация кэширования и индексов СУБД. Обнаружены 2 критические уязвимости (IDOR в API), устраненные за 48 часов. UI-тестирование выявило проблемы отображения на мобильных устройствах, решенные доработкой адаптивного дизайна.

2.5 Сравнение с аналогами

Power BI и Tableau обеспечивают мощную аналитику, но их стоимость в 6 раз выше разработанной платформы. Google Analytics бесплатен, но не поддерживает офлайн-данные и интеграцию с ERP, что реализовано в платформе. SaaS-решения (Looker) ограничены в кастомизации, тогда как платформа предлагает гибкую интеграцию с любыми источниками данных.

2.5.1 Влияние на бизнес-процессы

Внедрение платформы повысило точность решений за счет прогнозирования потребностей в товарных запасах; оптимизировало работу с данными через централизацию информации из CRM и ERP; сократило операционные затраты

на 37% за счет автоматизации отчетности; ускорило принятие решений через доступ к аналитике в реальном времени.

2.6 Внедрение и перспективы

2.6.1 Внедрение

Процесс включает этапы: подготовка (анализ источников данных: CRM, ERP, CSV); интеграция (настройка API для внешних систем); кастомизация (создание отчетов, настройка RBAC); обучение сотрудников; тестовый запуск и мониторинг.

2.6.2 Перспективы

Ключевые направления развития: внедрение ML-моделей для прогнозирования спроса и динамического ценообразования; расширение визуализации через интеграцию D3.js и картографических инструментов; разработка мобильных приложений (iOS/Android); генерация AI-рекомендаций по оптимизации ассортимента; интернационализация с поддержкой мультиязычности и валют.