

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики и информационных технологий

**ОПТИМИЗАЦИЯ ФАКТОРИНГОВОЙ СИСТЕМЫ: МАШИННОЕ
ОБУЧЕНИЕ, МИКРОСЕРВИСЫ, ЧИТАЕМЫЙ КОД И
АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 271 группы

Направление 09.04.01 — Информатика и вычислительная техника
факультета КНиИТ

Бурчуладзе Левана Александровича

Научный руководитель

профессор

Л.В. Кальянов

Заведующий кафедрой

доцент, к.ф.-м.н

Л.Б. Тяпаев

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы.

Современный бизнес всё больше зависит от технологических решений, которые позволяют оптимизировать процессы, повышать эффективность и снижать издержки. В сфере факторинга, где ключевую роль играют точность прогнозирования, скорость обработки данных и надёжность систем, внедрение современных технологий становится необходимым условием для конкурентоспособности. Оптимизация факторинговой системы требует комплексного подхода, включающего не только улучшение бизнес-процессов, но и модернизацию технической инфраструктуры.

Одним из ключевых направлений такой модернизации является переход от монолитной архитектуры к микросервисной. Микросервисы позволяют повысить гибкость системы, упростить поддержку и масштабировать, а также ускорить внедрение новых функций. Однако такой переход требует тщательной проработки архитектуры, улучшения читаемости кода и внедрения автоматизированного тестирования для обеспечения стабильности и качества продукта.

Ещё одним важным аспектом оптимизации является использование машинного обучения для анализа данных и прогнозирования. Машинное обучение позволяет выявлять скрытые закономерности в данных, автоматизировать процессы принятия решений и повышать точность прогнозов, что особенно важно в факторинге, где ключевыми являются оценка рисков и управление финансовыми потоками.

Автоматизированное тестирование играет crucial роль в обеспечении качества и надёжности системы. Оно позволяет выявлять ошибки на ранних этапах разработки, минимизировать риски при внедрении новых функций и обеспечивать стабильную работу системы под нагрузкой. Внедрение практик тестирования, таких как модульное, интеграционное и нагрузочное тестирование, способствует повышению устойчивости системы и снижению затрат на поддержку.

Цель магистерской работы – оптимизация факторинговой системы за счёт перехода к микросервисной архитектуре, улучшения читаемости кода, внедрения машинного обучения для анализа данных и автоматизированного тестирования для повышения качества продукта.

Для достижения данной цели были поставлены следующие задачи:

1. Изучить применение машинного обучения для анализа данных и оптимизации процессов факторинга.
2. Исследовать возможности перехода от монолитной к микросервисной архитектуре
3. Разработать подходы к улучшению читаемости кода.
4. Внедрить автоматизированное тестирование (модульное, интеграционное и нагрузочное) для повышения качества и надёжности системы.

Теоретическая значимость работы заключается в систематизации и развитии методов оптимизации систем за счет интеграции современных технологий, таких как машинное обучение, микросервисная архитектура и автоматизированное тестирование. Работа дополняет существующие исследования в области анализа данных для оценки рисков, а также предлагает новые подходы к проектированию и рефакторингу сложных финансовых систем.

Практическая значимость работы подтверждается внедрением разработанных решений в реальную факторинговую систему. Результаты, включая повышение производительности на 300% при обработке документов, улучшение масштабируемости и надежности системы, а также снижение рисков за счет прогнозирования банкротств ($AUC = 0.9375$), демонстрируют эффективность предложенных методов. Внедрение модульного и интеграционного тестирования обеспечило повышение качества кода и устойчивости системы, что имеет существенное значение для финансового сектора.

Структура и объём работы. Магистерская работа состоит из введения, 4 разделов, заключения, списка использованных источников и 4 приложений.

Общий объем работы – 67 страниц, из них 58 страниц – основное содержание, включая 18 рисунков, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Факторинг: что это и для чего используется?» В этом разделе автор подробно раскрывает суть факторинга как финансового инструмента, который помогает компаниям управлять дебиторской задолженностью и улучшать ликвидность. Основные моменты:

Определение факторинга: Это комплексный инструмент, позволяющий компаниям получать финансирование под уступку денежных требований. Пример: поставщик электроники с отсрочкой платежа 90 дней использует факторинг для покрытия текущих расходов.

Виды факторинга:

- С правом регресса: Риск неоплаты лежит на продавце.
- Без права регресса: Риск берет на себя фактор (например, банк).
- Агентский факторинг: Покупатель финансирует своих поставщиков.
- Открытый и закрытый: Отличаются уровнем прозрачности для дебиторов.

Преимущества:

- Улучшение ликвидности.
- Контроль сроков погашения задолженности.
- Снижение рисков неплатежей.

Вывод: Факторинг особенно полезен для компаний, работающих с отсрочкой платежа, так как предотвращает кассовые разрывы и способствует стабильности бизнеса.

Второй раздел «Машинное обучение» посвящен применению методов анализа данных и машинного обучения в факторинге. Были рассмотрены основы машинного обучения, включая линейную и логистическую регрессию, деревья решений и метод опорных векторов. Особое внимание

уделено анализу датасета СПАРК для прогнозирования банкротства компаний.

Датасет СПАРК:

- Содержит данные о банкротствах компаний за 2021–2023 гг.
- Объем: 600 тыс. строк, 136 столбцов.
- Целевая переменная: default (индикатор банкротства/ликвидации).

Анализ данных:

- Обнаружен дисбаланс классов: 185 тыс. наблюдений для класса 0 (не банкрот) против 10 тыс. для класса 1 (банкрот).
- Для балансировки данных использованы методы обработки и удаления выбросов.

Разработана модель логистической регрессии с $AUC = 0.9375$, что подтвердило её эффективность для оценки рисков.

Выводы: машинное обучение позволяет повысить точность прогнозирования в факторинговых операциях.

Третий раздел «Разработка» посвящен переходу от монолитной к микросервисной архитектуре и улучшению структуры проекта.

Проект был реорганизован с разделением на две основные директории:

- src — содержит исходный код приложения.
- test — включает модульные и интеграционные тесты.

Проведен рефакторинг файловой системы, внедрен Feature Sliced Design (FSD) для организации кода.

Особое внимание уделено разработке и интеграции новых микросервисов: электронной подписи, генерации отчетов, генерации водяных знаков и сервису единой авторизации и аутентификации.

Для обеспечения взаимодействия между сервисами использованы:

- REST API — для синхронных операций.
- Apache Kafka — для асинхронной обработки событий.

Описаны преимущества микросервисной архитектуры, включая гибкость и масштабируемость.

Переход на микросервисную архитектуру и оптимизация процессов обработки данных позволили достичь значительных улучшений в производительности, масштабируемости и эффективности разработки.

Ключевые результаты:

- Ускорение разработки печатных форм:
 - Время на модификацию существующих отчетов сократилось в 2,7 раза (с 16 до 6 чел./часов).
 - Время на создание новых отчетов уменьшилось в 2 раза (с 24 до 12 чел./часов).
- Рост производительности критичных операций:
 - Скорость подписания документов увеличилась на 300% (с ~500 до ~1500 операций в час).
 - Генерация водяных знаков также ускорилась в 3 раза (с ~500 до ~1500 операций в час).
- Оптимизация хранения данных:
 - Объем базы данных сократился на 75%, что значительно уменьшило нагрузку на инфраструктуру.
 - Время резервного копирования сократилось с более чем суток до 2–3 часов.
- Повышение масштабируемости и гибкости системы:
 - Независимое масштабирование сервисов позволило эффективно распределять ресурсы под пиковые нагрузки.

- Асинхронная обработка через Kafka устранила блокировки UI при массовых операциях.
- Упростилось внедрение новых функций и обновлений благодаря декомпозиции системы.

Так же была проведена оптимизация фронтенд-части факторинговой системы, включающая следующие ключевые изменения:

Переход от монолитной к микросервисной архитектуре:

- Фронтенд был разделён на три отдельных приложения для разных групп пользователей: клиенты, дебиторы и менеджеры.
- Для клиентов создан Единый Личный Кабинет Клиента (ЕЛКК) с использованием API-GATEWAY.
- Для дебиторов выделено отдельное фронтенд-пространство `debtor.rowi.com`.
- Для менеджеров создано внутреннее пространство `factoring-manager.rowi.com`, доступное только из корпоративной сети.

Повышение безопасности:

- Разделение пользовательских интерфейсов минимизировало риски утечки данных.
- Для менеджеров внедрена аутентификация через Keycloak-Internal, что упростило управление учётными записями.

Улучшение поддержки и разработки:

- Устранены зависимости между модулями, что ускорило внедрение изменений.
- Упрощён код за счёт удаления избыточного функционала.

Выводы: переход на микросервисы и улучшение структуры кода повысили эффективность и поддерживаемость системы.

Четвертый раздел «Тестирование» посвящен внедрению автоматизированного тестирования для повышения качества продукта. Тестирование программного обеспечения является критически важным этапом разработки, направленным на обеспечение качества, надежности и стабильности системы. В данном разделе рассматриваются ключевые виды тестирования, примененные в проекте:

Нагрузочное тестирование позволило оценить производительность системы при различных уровнях нагрузки, выявить узкие места и оптимизировать работу компонентов. Были проверены сценарии с высокой нагрузкой, что подтвердило устойчивость системы в условиях реальной эксплуатации.

Модульное тестирование обеспечило проверку корректности работы отдельных модулей в изоляции. На примере генерации паролей продемонстрирована эффективность использования фреймворка Xunit для создания и выполнения тестов, что повысило надежность кода.

Интеграционное тестирование подтвердило корректность взаимодействия между компонентами системы. С помощью инструментов, таких как FluentAssertions и Moq, были протестированы сценарии работы API, включая создание компаний, что гарантировало согласованность работы всех модулей.

Внедрение практики тестирования в проект позволило систематизировать процессы контроля качества, минимизировать риски ошибок и обеспечить стабильность системы. Реализованные тесты охватили как отдельные функции, так и комплексные сценарии, что способствовало повышению доверия к продукту.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной магистерской работы была проведена комплексная оптимизация факторинговой системы, включающая внедрение современных технологий и методологий разработки. Основной акцент был сделан на переход от монолитной архитектуры к микросервисной, что позволило значительно повысить гибкость, масштабируемость и надежность системы.

Были исследованы и применены методы анализа данных для прогнозирования банкротства и ликвидации компаний. Использование логистической регрессии показало высокую эффективность ($AUC = 0.9375$), что подтвердило возможность применения машинного обучения для оценки рисков в факторинге.

Разработана и внедрена новая архитектура, включающая специализированные сервисы (электронной подписи, генерации отчетов, работы с файлами и др.). Это позволило улучшить производительность, упростить поддержку и обеспечить независимое масштабирование компонентов.

Проведен рефакторинг файловой структуры проекта с использованием принципов Feature Sliced Design (FSD), что улучшило организацию кода и упростило его понимание для разработчиков.

Внедрены модульные и интеграционные тесты, что повысило качество кода и снизило риски возникновения ошибок. Особое внимание уделено генерации паролей, для которой были разработаны и успешно протестированы соответствующие модули.

Реализованные решения позволили достичь поставленных целей: повысить эффективность работы системы, снизить издержки на поддержку и обеспечить стабильность при высоких нагрузках. Дальнейшее развитие проекта может включать углубленное использование машинного обучения

для анализа данных, расширение функционала микросервисов и дальнейшую автоматизацию процессов тестирования.

Проделанная работа демонстрирует, что современные технологии и подходы к разработке играют ключевую роль в оптимизации финансовых систем, обеспечивая их конкурентоспособность и устойчивость в условиях динамично развивающегося бизнес-ландшафта.

Отдельные части магистерской работы были представлены на студенческой научной конференции 24.04.2025 в очном формате на факультете КНИиТ.

Основные источники информации:

1. Stenvall, J., Syväjärvi, A. Data mining in public and private sectors. New York: Information Science Reference, 2010.
2. Рашка С. Python и машинное обучение. Москва: ДМК Пресс, 2017.
3. Кузнецов, М.А. От монолита к микросервисам: стратегии миграции. — М.: ДМК Пресс, 2022.
4. Ричардсон, К. Микросервисы: Паттерны для масштабируемых и надежных приложений. — М.: ДМК Пресс, 2019.
5. Расолько, А.М., Тонкович, И.Н. Автоматизация функционального тестирования веб-приложений с использованием фреймворка // Актуальные проблемы науки ххi века, № 7, 2018. С. 19-25.