Научная статья
УДК 004.8:004.9
https://doi.org/10.17586/2713-1874-2025-4-83-100

РАЗРАБОТКА ИНТЕГРИРОВАННОЙ ПАНЕЛИ УПРАВЛЕНИЯ ЖИЗНЕННЫМ ЦИКЛОМ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ИТ-ОРГАНИЗАЦИЙ С РАСПРЕДЕЛЕННОЙ ИНФРАСТРУКТУРОЙ

Наталья Викторовна Добренко[1], Виктория Юрьевна Бабан[2,3]✉, Анастасия Вадимовна Чернышева[4], Антон Игоревич Говоров[5]

[1,2,4,5]Университет ИТМО, Санкт-Петербург, Россия
[3]ООО «УНО-СОФТ», Санкт-Петербург, Россия
[1]dobrenkonv@itmo.ru, https://orcid.org/0000-0001-6206-8033
[2,3]viktoriya.baban.03@mail.ru✉
[4]avchernysheva@itmo.ru, https://orcid.org/0000-0002-9956-6607
[5]antongovorov@gmail.com, https://orcid.org/0009-0005-6674-1666
Язык статьи – русский

Аннотация: В статье рассмотрена актуальная проблема инструментальной фрагментации в MLOps-процессах, возникающая в организациях, использующих распределённую инфраструктуру и набор отдельных инструментов для работы с моделями машинного обучения. К таким организационным системам относятся компании, разрабатывающие ML-сервисы, и аналитические подразделения, применяющие машинное обучение для автоматизации бизнес-процессов. Целью исследования является разработка научно обоснованных требований к созданию интегрированной панели управления, предназначенной для централизации операций, унификации рабочих процессов и снижения операционных затрат. В работе применен комплексный методологический подход: сравнительный анализ пяти репрезентативных платформ с использованием взвешенных критериев и анкетирование специалистов различных профилей. На основе синтеза результатов сформирован и приоритизирован по методу MoSCoW комплекс требований, а также определены ключевые архитектурные принципы, включая микросервисную архитектуру, API-first подход и роле-ориентированный интерфейс. Путем построения дискретно-событийной модели типа M/M/m проведена верификация проектных решений и количественно оценены ключевые метрики производительности. Результатом исследования является научно обоснованная основа для разработки прототипа унифицированной панели управления, позволяющая преодолеть разрыв между теоретическими принципами MLOps и их инструментальной реализацией.

Ключевые слова: имитационное моделирование, интеграционная платформа, машинное обучение, микросервисная архитектура, панель управления, пользовательский интерфейс, производительность системы, системы массового обслуживания, требования к программному обеспечению, управление жизненным циклом ML, MLOps

Ссылка для цитирования: Добренко Н. В., Бабан В. Ю., Чернышева А. В., Говоров А. И. Разработка интегрированной панели управления жизненным циклом моделей машинного обучения для ИТ-организаций с распределенной инфраструктурой // Экономика. Право. Инновации. – 2025. – Т. 13. – № 4. – С. 83–100. – https://doi.org/10.17586/2713-1874-2025-4-83-100

DEVELOPMENT OF AN INTEGRATED LIFECYCLE MANAGEMENT PANEL FOR MACHINE LEARNING MODELS FOR IT ORGANIZATIONS WITH DISTRIBUTED INFRASTRUCTURE

Natalya V. Dobrenko[1], Viktoriia Iu. Baban[2,3]✉, Anastasiia V. Chernysheva[4], Anton I. Govorov[5]

[1,2,4,5]ITMO University, Saint Petersburg, Russia
[3]JUNO-SOFT LLC, Saint Petersburg, Russia
[1]dobrenkonv@itmo.ru, https://orcid.org/0000-0001-6206-8033
[2,3]viktoriya.baban.03@mail.ru✉
[4]avchernysheva@itmo.ru, https://orcid.org/0000-0002-9956-6607
[5]antongovorov@gmail.com, https://orcid.org/0009-0005-6674-1666
Article in Russian

Abstract: The article addresses the pressing issue of tool fragmentation in MLOps processes, which arises in organizations that operate using distributed infrastructure and a set of disparate tools for working with machine learning models. Such organizational systems include companies that develop ML services and internal analytics departments that apply machine learning to automate business processes. The aim of the study is to develop scientifically grounded requirements for creating an integrated control panel designed to centralize operations, unify workflows, and reduce operational costs. A comprehensive methodological approach was applied: a comparative analysis of five representative platforms using weighted criteria and a survey of specialists from various profiles. Based on a synthesis of the results, a set of requirements was formed and prioritized using the MoSCoW method, and key architectural principles were defined, including a microservice architecture, an API-first approach, and a role-oriented interface. By constructing a discrete-event M/M/m type model, the design solutions were verified and key performance metrics were quantitatively assessed. The result of the research is a scientifically substantiated foundation for developing a prototype unified control panel, which helps bridge the gap between the theoretical principles of MLOps and their tool implementation.

Keywords: simulation modeling, integration platform, machine learning, microservice architecture, control panel, user interface, system performance, queuing systems, software requirements, ML lifecycle management, MLOps

For citation: Dobrenko N. V., Baban V. Iu., Chernysheva A. V., Govorov A. I. Development of an Integrated Lifecycle Management Panel for Machine Learning Models for IT Organizations with Distributed Infrastructure. *Ekonomika. Pravo. Innovacii.* 2025. Vol. 13. No. 4. pp. 83–100. (In Russ.). https://doi.org/10.17586/2713-1874-2025-4-83-100

Введение. Машинное обучение (МО) играет центральную роль в цифровой трансформации различных сфер – от медицины и финансов до промышленности и торговли. Согласно отчетам аналитиков, мировой рынок МО демонстрирует стремительный рост: ожидается, что к 2032 г. он достигнет объема около $309,68 млрд долларов США при среднегодовом темпе роста примерно 30,5% [1].

В современных условиях разработка и внедрение моделей машинного обучения представляет собой сложный многоэтапный процесс, включающий сбор и подготовку данных, построение признаков, обучение моделей, проверку их качества, развертывание и последующий мониторинг в рабочей среде. Каждый из этих этапов требует использования специализированных инструментов: систем управления версиями кода (GitLab, GitHub), платформ управления экспериментами (MLflow, Neptune.ai), инструментов мониторинга (Grafana, Prometheus), систем контейнеризации и оркестрации (Docker, Kubernetes).

Такая инструментальная фрагментация порождает ряд серьезных проблем организационного и технического характера. Специалисты вынуждены переключаться между различными интерфейсами, что увеличивает когнитивную нагрузку и риск ошибок. Отсутствие единого представления о состоянии проекта МО затрудняет координацию работы команды и принятие решений.

Существующие коммерческие платформы (Amazon SageMaker, Google Vertex AI, Yandex Cloud DataSphere) предлагают интегрированные решения «все-в-одном», но характеризуются рядом ограничений: высокая стоимость использования, привязка к конкретной облачной экосистеме, ограниченные возможности кастомизации. Открытые решения (Kubeflow, MLflow) предоставляют большую гибкость, но требуют значительных технических компетенций для развертывания, настройки и поддержки, что делает их малодоступными для команд без глубоких DevOps-навыков.

Актуальность исследования определяется растущей потребностью организаций в построении удобных и стабильных ML-конвейеров – процессов машинного обучения, охватывающих полный жизненный цикл модели. В компаниях с распределённой инфраструктурой и большим количеством прикладных задач МО специалисты сталкиваются с необходимостью работы в нескольких несвязных системах одновременно. В результате возникает фрагментация инструментов, усложняющая эксплуатацию моделей и увеличивающая затраты. Это усиливает спрос на единую панель управления, которая могла бы обеспечить упорядоченное взаимодействие между существующими компонентами инфраструктуры для машинного обучения (ML-инфраструктуры).

Постановка задачи и цель исследования. Современные коммерческие и технологические организации, использующие машинное обучение в своих продуктах или внутренних сервисах, сталкиваются с пробле-

мой эксплуатации моделей в условиях распределённой инфраструктуры. В таких организационных системах процессы МО охватывают несколько подразделений: разработку, аналитику, MLOps-поддержку, деплой моделей и мониторинг. Отсутствие единой точки управления приводит к росту операционных затрат, усложняет коммуникацию между командами и снижает устойчивость ML-конвейеров.

В этих условиях возникает необходимость в интегрированной панели управления, способной объединить разнородные инструменты (MLflow, Docker/Portainer, Grafana, GitLab и др.) и предоставить специалистам удобный интерфейс для выполнения повседневных операций — от запуска задач до анализа метрик. Такая панель особенно востребована:

– в организациях с гибридной или распределённой инфраструктурой, сочетающей локальные серверы организации и облачные ресурсы,

– в компаниях, развивающих собственные продукты МО,

– в командах, где различные роли используют разные инструменты,

– в ситуациях, когда требуется централизованный контроль за состоянием моделей и задач.

Цель исследования – определить требования и архитектурные принципы построения интегрированной панели управления для задач машинного обучения, предназначенной для применения в компаниях с распределённой ML-инфраструктурой и используемой для унификации рабочих процессов, повышения стабильности ML-конвейеров и сокращения операционных затрат.

Для достижения цели в рамках исследования решаются следующие задачи:

1) анализ целевой аудитории и выявление потребностей ключевых ролей в процессе работы с ML-инфраструктурой,

2) сравнительный анализ интерфейсов и функциональных возможностей существующих интегрированных платформ МО (ML-платформ) с целью выявления их преимуществ и ограничений,

3) формирование и приоритизирование комплекса функциональных и нефункциональных требований к разрабатываемой панели управления с применением метода MoSCoW,

4) определение архитектурных принципов построения системы, обеспечивающих гибкость, масштабируемость и удобство интеграции,

5) верификация проектных решений путем построения имитационной модели и оценки производительности системы при различных сценариях нагрузки.

Литературный обзор. В последние годы машинное обучение активно развивается и становится неотъемлемой частью цифровой трансформации различных отраслей. Появление большого количества инструментов, библиотек и платформ сделало процесс построения и внедрения моделей доступнее, но вместе с тем — сложнее в управлении. Это привело к формированию отдельного направления, получившего название MLOps, которое объединяет инженерные и организационные практики для автоматизации жизненного цикла моделей машинного обучения [2].

Теоретической основой современных процессов разработки моделей остаётся кросс-отраслевой стандартный процесс для интеллектуального анализа данных (Cross-Industry Standard Process for Data Mining, CRISP-DM), который описывает итеративный процесс от определения бизнес-задачи до развертывания и сопровождения модели [3, 4]. В современных условиях эта методология была адаптирована для задач MLOps с акцентом на автоматизацию и мониторинг. Как показывают исследования в области управления артефактами МО, эффективное управление жизненным циклом требует системного подхода к версионированию и отслеживанию всех компонентов систем МО [5].

Значительное внимание в научных и прикладных работах уделяется анализу инструментов, обеспечивающих поддержку различных этапов жизненного цикла моделей машинного обучения. Так, решения типа MLflow ориентированы на управление экспериментами — хранение параметров, метрик и артефактов обучения. Для наблюдения за состоянием инфраструктуры и производительностью моделей в эксплуатационной среде применяются системы визуального мониторинга, такие как Grafana. Управление контейнеризованными сервисами и их развертыва-

нием осуществляется при помощи инструментов Docker и Portainer. Однако при совместном использовании этих решений возникает потребность в согласованной интеграции, которая обеспечивает целостность MLOps-процессов.

В научных публикациях и отраслевых источниках рассматриваются различные подходы к объединению инструментов, среди которых наиболее распространённым является взаимодействие через API [6, 7]. Такой подход обеспечивает гибкость и масштабируемость, позволяя адаптировать процессы под конкретные требования проекта. В то же время альтернативные решения – использование готовых коннекторов и шин данных [8] – применяются реже из-за ограниченной универсальности и зависимости от конкретных технологий.

Несмотря на большое количество публикаций, посвящённых отдельным аспектам MLOps, в научной литературе по-прежнему недостаточно исследований, предлагающих целостные архитектурные решения, объединяющие лучшие практики управления проектами МО в едином интерфейсе, адаптированном под различные роли пользователей. Большинство работ сосредоточено на методологических и организационных вопросах — улучшении коллаборации команд, стандартизации процессов и интеграции с DevOps [5, 9], — в то время как практические инструменты для реализации этих подходов описаны значительно реже, что сохраняет разрыв между теорией и применением.

Существующие коммерческие платформы, такие как Amazon SageMaker и Google Vertex AI, предлагают интегрированные решения, но характеризуются привязкой к конкретным облачным экосистемам и ограниченными возможностями кастомизации [10]. В свою очередь, открытые решения, например Kubeflow, предоставляют большую гибкость, но требуют значительных технических компетенций для развертывания и поддержки.

Таким образом, анализ литературы показывает, что существующие решения имеют как сильные стороны, так и существенные ограничения, влияющие на удобство и эффективность их применения в реальных условиях. Это подтверждает необходимость разработки интегрированной панели управления, объединяющей преимущества различных подходов и учитывающей особенности инфраструктуры и рабочих процессов организаций.

Материалы и методы исследования. Для формирования обоснованных требований к интерфейсу панели управления задачами машинного обучения в работе был применен комплексный подход, сочетающий сравнительный анализ существующих решений и исследование потребностей целевой аудитории. Такой метод позволил, с одной стороны, учесть лучшие отраслевые практики, а с другой — обеспечить соответствие системы реальным рабочим процессам специалистов.

Для выявления лучших практик, ограничений и ключевых характеристик существующих решений был проведен сравнительный анализ пяти репрезентативных ML-платформ: Amazon SageMaker, Google Cloud Vertex AI, Yandex Cloud DataSphere, Apache Ambari и Kubeflow Central Dashboard. Отбор платформ осуществлялся по критериям востребованности в профессиональном сообществе, полноты реализации жизненного цикла моделей машинного обучения, наличия развитого веб-интерфейса и активной поддержки со стороны разработчиков. Объективность оценки обеспечивала система показателей, учитывающая функциональные, пользовательские, технические и интеграционные аспекты, а итоговое сравнение проводилось с применением метода интегральной оценки.

Важным элементом исследования стало проведение анкетирования среди специалистов по машинному обучению. Цель опроса заключалась в выявлении потребностей, затруднений и предпочтений пользователей при работе с существующими инструментами управления задачами машинного обучения.
Анкета была разработана на основе анализа научных публикаций и практических обзоров существующих платформ (включая MLflow, Kubeflow, Yandex DataSphere и Ambari). Для формирования вопросов применялся метод контент-анализа [11]: изучались функциональные возможности, структура интерфейсов и сценарии использования указанных систем. На основе этого были зафиксированы наиболее часто упоминаемые трудности и

ожидания специалистов, связанные с повседневным использованием инструментов МО, и по ним сформирована структура анкеты. Она включала как закрытые, так и открытые вопросы, что позволило сочетать количественную и качественную оценку пользовательского опыта.

Финальная версия анкеты была доработана по результатам обсуждения с заказчиком и другими специалистами, знакомыми с практиками MLOps. Их комментарии помогли уточнить формулировки вопросов и исключить дублирующиеся пункты, обеспечив логичность и практическую направленность анкеты.

Для обработки полученных данных применялись методы тематического анализа [12] для качественных ответов и статистический метод [13] для количественных данных, что позволило оценить распространённость отдельных проблем и выявить наиболее значимые для специалистов аспекты работы с платформами машинного обучения. На заключительном этапе исследования для структурирования требований к разрабатываемой системе был использован метод приоритизации MoSCoW [14], позволяющий выделить наиболее значимые функциональные возможности проектируемой панели управления.

Для верификации проектных решений и количественной оценки производительности проектируемой системы был применен метод дискретно-событийного имитационного моделирования [15, 16]. Данный метод позволяет воспроизвести поведение сложной системы во времени, учитывая стохастический характер поступления и обработки задач. Моделирование было реализовано на языке Python с использованием специализированной библиотеки SimPy. Для обеспечения статистической достоверности результатов применялся метод Монте-Карло с многократными прогонами модели при различных начальных условиях [16].

Проведение опроса. Исследование проводилось в два этапа, что позволило охватить как широкое профессиональное сообщество, так и сфокусироваться на целевой группе, для которой планируется внедрение разрабатываемой панели управления.

Первый этап опроса был нацелен на формирование общего понимания проблем и практик в индустрии. В нем приняли участие около 100 респондентов – специалистов по машинному обучению и MLOps, рекрутированных через профессиональные сообщества и социальные сети. Выборка была сформирована с учетом профессиональных ролей и опыта работы респондентов.

Второй этап опроса был сконцентрирован на глубоком изучении рабочих процессов и потребностей конкретной команды компании-заказчика (20 человек), что позволило получить детализированные требования в контексте реальных бизнес-процессов.

Анкета включала вопросы, направленные на выявление демографических данных, рабочих практик и функциональных предпочтений, например: «Какова ваша основная роль в команде?», «Какие этапы жизненного цикла моделей вы обычно выполняете?», «Какие ключевые инструменты вы используете ежедневно для выполнения полного цикла задачи (от эксперимента до продакшена)?», «Какие проблемы и неудобства вы испытываете при работе с текущими инструментами?», «Используете ли вы интегрированные платформы?». Опрос предлагал отметить наиболее полезные и необходимые функции панели управления, а также оставить в свободной форме комментарий/пожелание.

Результаты опроса. Проведенный опрос подтвердил наличие существенной дезинтеграции инструментальной среды, используемой в рабочих процессах, связанных с машинным обучением. Подавляющее число участников исследования (95%) применяют Jupyter Notebook в качестве основного инструмента, однако в 65% случаев его использование сочетается с Docker/Portainer, а в 45% – с MLflow. Полученные данные свидетельствуют о том, что Jupyter служит преимущественно для исследовательских задач и создания прототипов, не покрывая полный цикл операций MLOps. Готовые MLOps-платформы нашли применение лишь у 17% респондентов, при этом в 60% случаев в качестве сдерживающего фактора была указана высокая стоимость, а в 40% – недостаточная адаптивность под специфические рабочие процессы.

Статистический анализ выявил достоверные различия в запросах среди специалистов различного профиля, что наглядно представлено на рисунке 1.

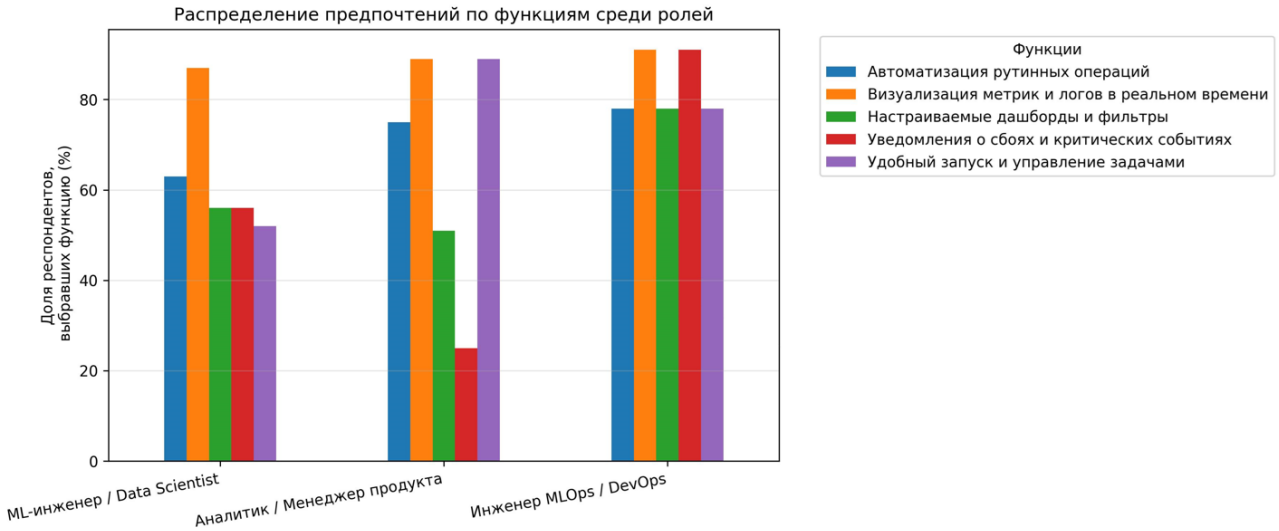

Рисунок 1 – Распределение приоритетов функций интерфейса среди различных профессиональных групп

Источник: составлено авторами

Результаты показывают четкую профессиональную специализацию, которая в ряде случаев отражает совмещение ролей в реальных командах:

– группа «ML-инженер/Data Scientist» проявляет наибольший интерес к функциям, связанным с визуализацией метрик и логов в реальном времени и автоматизации рутинных задач, что соответствует фокусу на исследовательском цикле и индустриализации моделей;

– группа «Инженер MLOps/DevOps» выделяет в качестве критически важных функций визуализацию метрик и логов, а также уведомления о сбоях, что отражает ответственность за надежность и эффективность инфраструктуры;

– группа «Аналитик/Менеджер продукта» представляет бизнес-сторону и заинтересована в визуализации данных и удобстве интерфейса.

Для целей проектирования архитектуры, направленной на решение проблем технических специалистов, фокус был сделан на первых двух группах.

Ответы на вопрос «Какие данные и метрики вы чаще всего используете для анализа и контроля?» позволили детализировать эти различия на уровне информационных потребностей:

– специалисты, относящие себя к области Data Science и разработки моделей, в 85% случаев указывали метрики качества модели (accuracy, precision, recall, F1-score, AUC-ROC) и визуализации процесса обучения. Для них критически важны артефакты экспериментов (сохранённые веса модели, графики) и параметры запуска (гиперпараметры, версия датасета);

– специалисты, идентифицирующие себя с MLOps и инфраструктурой, в 79% случаев называли системные метрики (загрузка CPU/GPU, потребление памяти, сетевой трафик) и операционные показатели (latency и throughput инференс-сервисов, частота ошибок). Для них ключевыми являются логи оркестраторов (Kubernetes, Airflow) и статус пайплайнов.

Качественный анализ текстовых ответов респондентов позволил идентифицировать ключевые проблемные области:

– необходимость постоянного переключения между множеством интерфейсов, что приводит к значительным временным затратам;

– затрудненность диагностики возникающих проблем из-за распределенности журналов выполнения и метрик по различным системам;

– отсутствие консолидированного представления о статусе выполняемых экспериментов и конвейеров обработки данных.

Данные проблемы согласуются с результатами других исследований, где респонденты также отмечали сложности, вызванные избыточной бюрократией, жесткими процессами и обилием инструментов, не интегрированных в единый рабочий процесс [17].

Исследование внутренней команды заказчика продемонстрировало высокий уровень зрелости используемых практик: 78% участников применяют MLflow для управления экспериментами, а 78% задействуют Airflow для оркестрации рабочих процессов. Наиболее значимыми функциями будущей системы были названы: управление задачами, визуализация метрик и журналов в реальном времени, настраиваемые информационные панели и система оповещений. Распределение приоритетов среди ключевых функций представлено на рисунке 2.

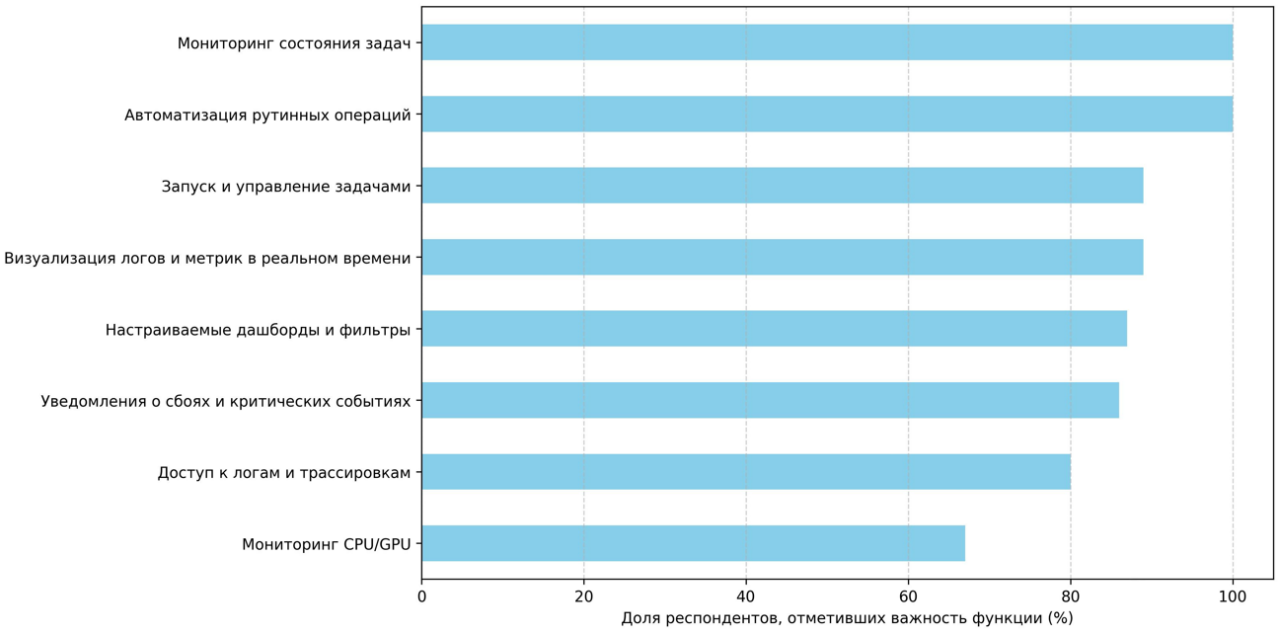

Рисунок 2 – Приоритеты ключевых функций панели управления по результатам опроса целевой команды

Источник: составлено авторами

Сравнительный анализ существующих платформ. Для выявления лучших практик и ограничений существующих решений был проведен сравнительный анализ пяти репрезентативных ML-платформ. Сравнение осуществлялось по методике интегральной оценки, позволяющей свести разнородные критерии в единую количественную метрику. Критерии оценки включили четыре аспекта, определяющих эффективность платформы:

1) функциональность – охват этапов жизненного цикла модели МО;

2) эргономика интерфейса – интуитивность и отзывчивость интерфейса, влияющие на скорость освоения платформы и эффективность работы пользователей;

3) технические параметры – стабильность работы, масштабируемость и отказоустойчивость архитектуры;

4) интеграционный потенциал – совместимость с существующей IT-инфраструктурой.

Для сравнения платформ был использован метод интегральной оценки, основанный на взвешенных показателях, где каждой группе критериев был присвоен весовой коэффициент, отражающий ее относительную важность (таблица 1).

Для расчета интегрального показателя использовалась формула:

$$I = \frac{\sum_{j=1}^{m} \sum_{i=1}^{n} I_{ij} * p_i}{m}, \quad (1)$$

где m – число экспертов, n – количество показателей, I_{ij} – оценка платформы по i-му критерию j-м экспертом, p_i – весовой коэффициент i-го критерия.

Таблица 1

Весовые коэффициенты групп критериев

Источник: составлено авторами

Группа критериев	Вес
Функциональные параметры	0,4
Эргономика интерфейса	0,3
Технические параметры	0,2
Интеграционный потенциал	0,1

Результаты экспертных оценок приведены в таблице 2, результаты расчетов интегральной оценки, отсортированные по убыванию – в таблице 3, значения оценки критериев первых трех приложений с самой высокой интегральной оценкой – в таблице 4.

Таблица 2

Оценки платформ экспертами

Источник: составлено авторами

Платформа	Эксперт 1	Эксперт 2	...	Эксперт m
Google Cloud Vertex AI	9,3	9,1	...	8,9
Amazon SageMaker	8,8	8,6	...	9,1
Kubeflow	7,8	7,6	...	7,9
Yandex Cloud DataSphere	6,6	6,4	...	6,5
Apache Ambari	5,9	5,7	...	6,0

Таблица 3

Интегральная оценка ML-платформ

Источник: составлено авторами

Платформа	Интегральная оценка (I)
Google Cloud Vertex AI	9,2
Amazon SageMaker	8,7
Kubeflow	7,7
Yandex Cloud DataSphere	6,5
Apache Ambari	5,8

Таблица 4

Оценка лидирующих платформ по группам критериев (по 10-балльной шкале)

Источник: составлено авторами

Группа критериев	Google Cloud Vertex AI	Amazon SageMaker	Kubeflow
Функциональные параметры	9,5	9,2	8,1

Группа критериев	Google Cloud Vertex AI	Amazon SageMaker	Kubeflow
Эргономика интерфейса	9,3	8,1	6,8
Технические параметры	8,8	8,6	8,4
Интеграционный потенциал	9,2	9,1	7,9

Результаты анализа показали, что коммерческие платформы (Google Cloud Vertex AI, Amazon SageMaker) демонстрируют высокий уровень зрелости и удобства, но обладают ограниченными возможностями кастомизации. Открытые решения (Kubeflow), напротив, предоставляют высокую гибкость, но требуют значительных усилий по настройке и сопровождению.

Формирование требований к панели управления. На основе синтеза результатов, полученных в ходе анализа потребностей пользователей и сравнительной оценки платформ, был сформирован комплекс требований к проектируемой системе. Требования структурированы по методу MoSCoW, что обеспечивает четкую приоритизацию функциональности для поэтапной реализации (таблица 5).

Таблица 5

Сводные требования к панели управления

Источник: составлено авторами

Приоритет	Категория требований	Ключевые функции
Must Have	Управление задачами	Централизованный запуск, мониторинг статуса и прогресса, управление очередями задач
	Интеграция с MLOps	Глубокая интеграция с MLflow, Grafana, Docker/Portainer
	Визуализация и мониторинг	Агрегированный просмотр логов и трассировок, система уведомлений о сбоях
	Пользовательский интерфейс	Ролевые шаблоны интерфейса, интуитивная навигация, время отклика ≤ 2 сек, адаптивный дизайн
Should Have	Автоматизация и оркестрация	Интеграция с GitLab CI/CD, автоматическое масштабирование ресурсов
	Управление ресурсами	Мониторинг использования CPU, GPU, памяти в реальном времени
	Кастомизация	Настраиваемые дашборды, персонализация рабочего пространства
Could Have	Дополнительная функциональность	Расширенное управление очередями, поддержка WCAG 2.1 AA, интеграция с внешними системами мониторинга

Сформированные требования обеспечивают баланс между функциональной полнотой, ориентированной на специфику машинного обучения, и высокими стандартами пользовательского опыта.

Архитектурные принципы. На основе анализа лучших практик существующих платформ и выявленных пользовательских требований были сформулированы ключевые архитектурные принципы построения системы. Основой проектируемого решения является микросервисная архитектура, предполагающая разделение системы на независимые сервисы для управления задачами, интеграции с MLflow, Grafana и другими инструментами. Такой подход обеспечивает гибкость, масштабируемость и устойчивость к отказам. Логическая структура компонентов системы и их взаимодействия представлена на рисунке 3.

Рисунок 3 – Диаграмма компонентов
Источник: составлено авторами

Взаимодействие с системой и между ее компонентами должно строиться на принципах API-first подхода. Проектирование согласованных и документированных API на начальном этапе является необходимым условием для обеспечения надежной интеграции как с внешними системами, так и между внутренними сервисами платформы. Последовательность взаимодействия компонентов в рамках типового сценария иллюстрирует рисунок 4.

Для определения чётких сценариев взаимодействия результаты опроса (рисунок 1) были интерпретированы в виде трёх ключевых ролей пользователей, охватывающих жизненный цикл модели:

– Data Scientist – фокус на анализе и экспериментировании;

– ML-инженер – фокус на индустриализации, развертывании и контроле качества моделей;

– MLOps-инженер – фокус на надёжности, оркестрации и мониторинге инфраструктуры.

Данная модель, дополненная ролью Администратора системы для управления платформой, легла в основу проектирования роле-ориентированного интерфейса. Диаграмма прецедентов (рисунок 5) иллюстрирует ключевые сценарии использования системы для каждой из формализованных ролей.

Реализация роле-ориентированного подхода также требует учёта потенциальных конфликтов интересов между ролями в рамках общего workflow, например:

– Data Scientist заинтересован в гибкости экспериментов, в то время как ML-инженер и MLOps-инженер – в стандартизации процессов для надёжной индустриализации;

– ML-инженер стремится к скорости развёртывания, а MLOps-инженер – к стабильности и контролю над инфраструктурой.

Проектирование интерфейса и процессов системы должно не просто отражать различия ролей, но и способствовать сглаживанию этих конфликтов через встроенные механизмы: шаблоны для стандартизации, контролируемые пайплайны деплоя с откатом, систему квот и утверждения запросов.

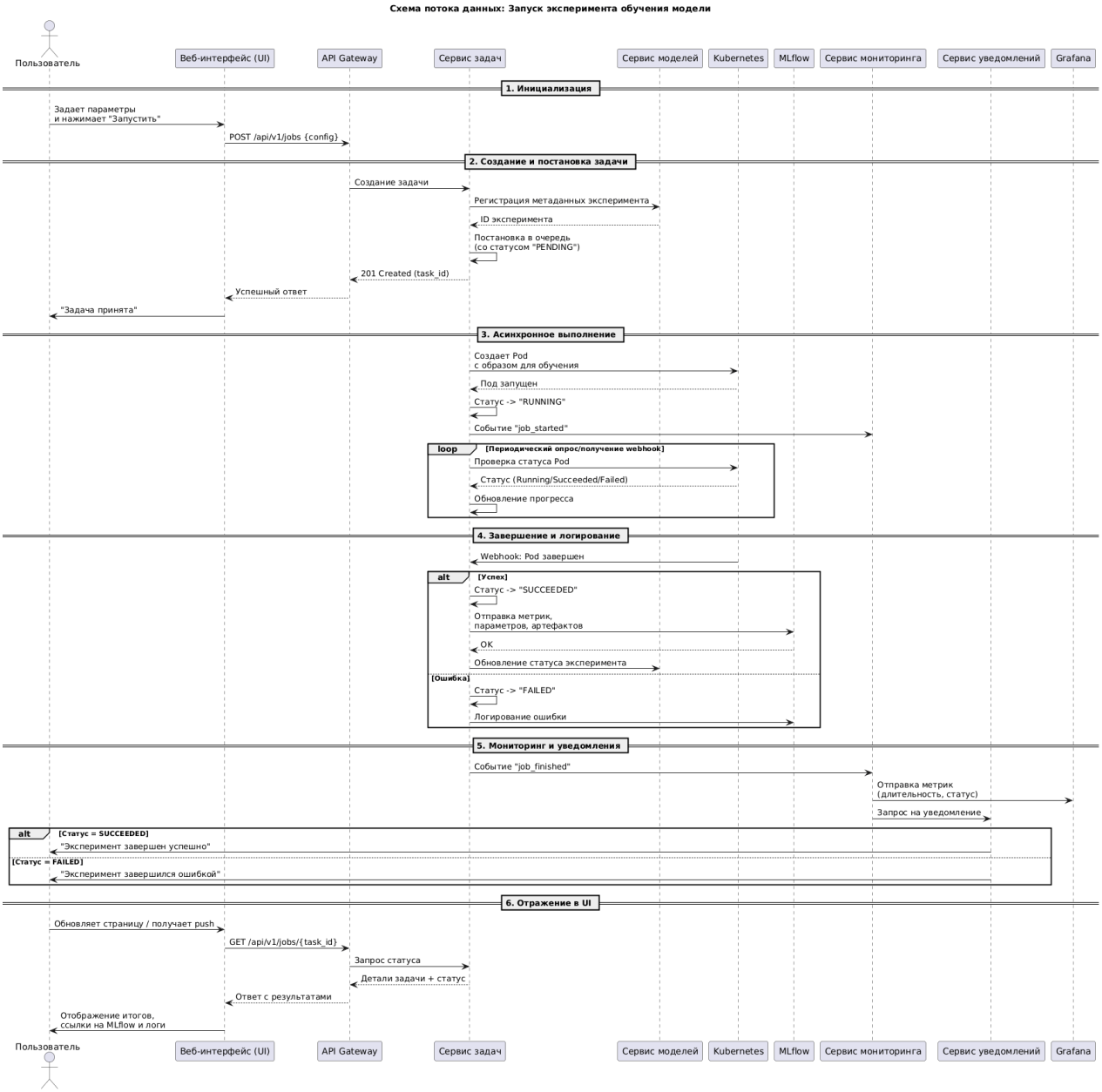

Рисунок 4 – Схема потока данных для сценария «Запуск эксперимента»
Источник: составлено авторами

Моделирование и оценка производительности. Целью проведенного имитационного эксперимента являлась количественная оценка ключевых метрик производительности ядра панели управления (время отклика системы, время ожидания в очереди, загрузка ресурсов сервисов) при различных сценариях нагрузки.

В основе модели лежит многоканальная система массового обслуживания типа M/M/m [18], что обусловлено характером нагрузки на API-шлюз и основные сервисы (сервис задач и сервис моделей): они обрабатывают короткие транзакционные запросы (создание задачи, проверка статуса, запрос метрик), время синхронной обработки которых может быть аппроксимировано экспоненциальным распределением.

Длительные асинхронные операции (запуск обучения в K8s, выполнение пайплайна в Airflow) выполняются внешними системами; панель управления лишь инициирует их и отслеживает статус через опрос или webhook, что не нагружает её вычислительные ресурсы в течение всего времени выполнения ML-задачи. Выбор многоканальной модели (m серверов) отражает архитектуру системы, рассчитанную на параллельное обслуживание множества пользователей.

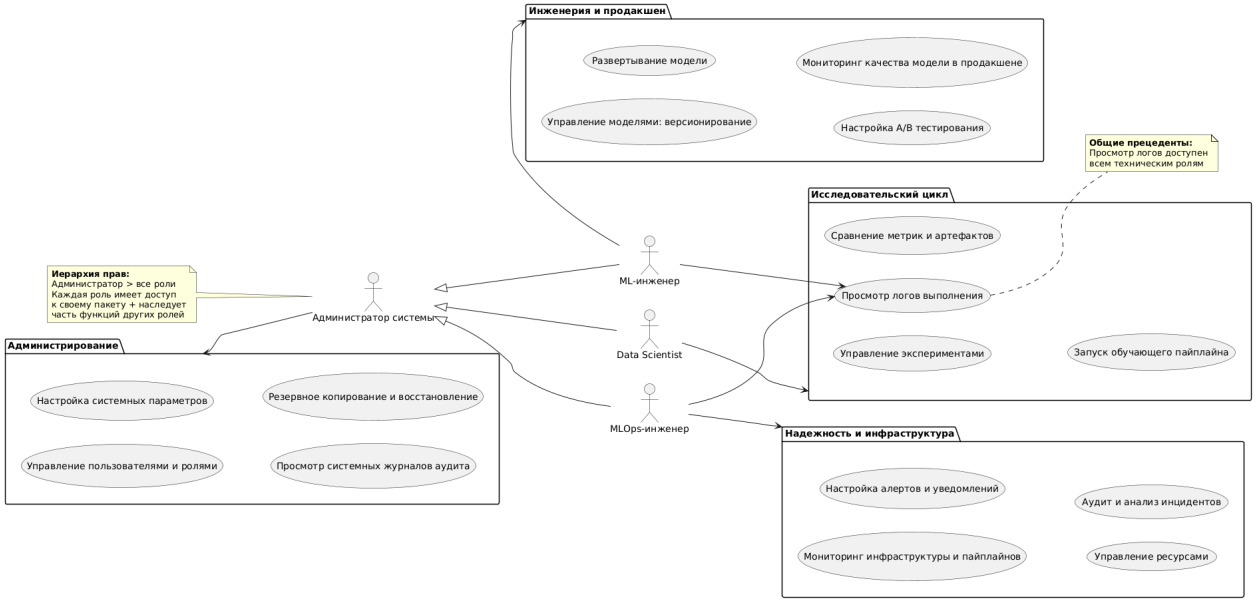

Рисунок 5 – Диаграмма прецедентов
Источник: составлено авторами

Данная модель предполагает пуассоновский поток поступающих задач с интенсивностью λ и экспоненциальное распределение времени обслуживания с параметром μ. Количество параллельных каналов обслуживания (серверов/агентов) обозначено как *c*.

Для расчета ключевых показателей эффективности использовались следующие формулы [15]:

– коэффициент загрузки ресурсов:

$$\rho = \frac{\lambda}{m * \mu} < 1, \quad (2)$$

– вероятность ожидания в очереди (формула Эрланга С):

$$P_W = \frac{\left[\frac{(m\rho)^m}{m!(1-\rho)}\right]}{\sum_{k=0}^{m-1}\frac{(m\rho)^k}{k!}+\frac{(m\rho)^m}{m!(1-\rho)}}, \quad (3)$$

– среднее время ожидания в очереди:

$$W_q = \frac{P_W}{m\mu - \lambda}, \quad (4)$$

– среднее время отклика задачи:

$$T_{resp} = W_q + \frac{1}{\mu}. \quad (5)$$

Для обеспечения репрезентативности результатов был применен метод Монте-Карло с 30 прогонами для каждого сценария. Период моделирования составил 168 часов (1 неделя). Были исследованы четыре сценария, варьирующие ключевые параметры системы (таблица 6).

Таблица 6

Параметры исследуемых сценариев имитационного эксперимента

Источник: составлено авторами

Сценарий	Количество серверов, m	Интенсивность входа, λ (задач/час)	Интенсивность обслуживания, μ (задач/час/сервер)
Базовая нагрузка	3	1,5	2,0
Высокая нагрузка	3	2,5	2,0
Недостаток ресурсов	2	2,0	2,0
Оптимизированная	4	2,0	2,5

Для наглядного представления базовой логики обработки запросов в системе на рисунке 6 приведена упрощённая блок-схема имитационной модели. Полная модель, реализованная на Python с использованием библиотеки SimPy, учитывала суточные колебания нагрузки (пики в часы 9–11 и 14–16, снижение ночью), случайные всплески спроса и потенциальные временные сбои серверов. Для комплексной оценки поведения системы под различной нагрузкой на рисунках 7–9 представлены диаграммы, иллюстрирующие распределение времени отклика, ожидания и загрузки ресурсов.

Сводные результаты эксперимента представлены в таблице 7.

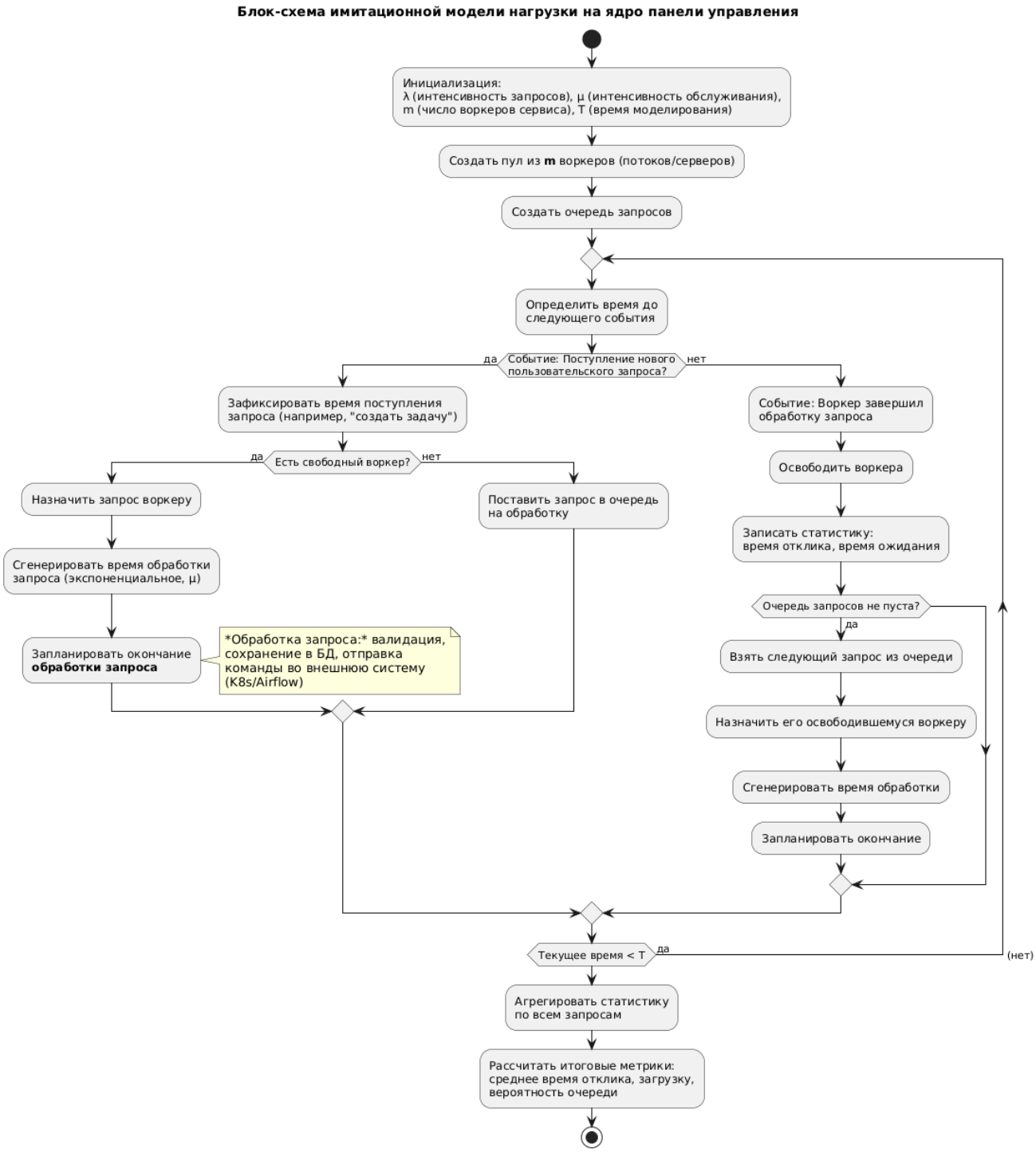

Рисунок 6 – Базовая блок-схема имитационной модели
Источник: составлено авторами

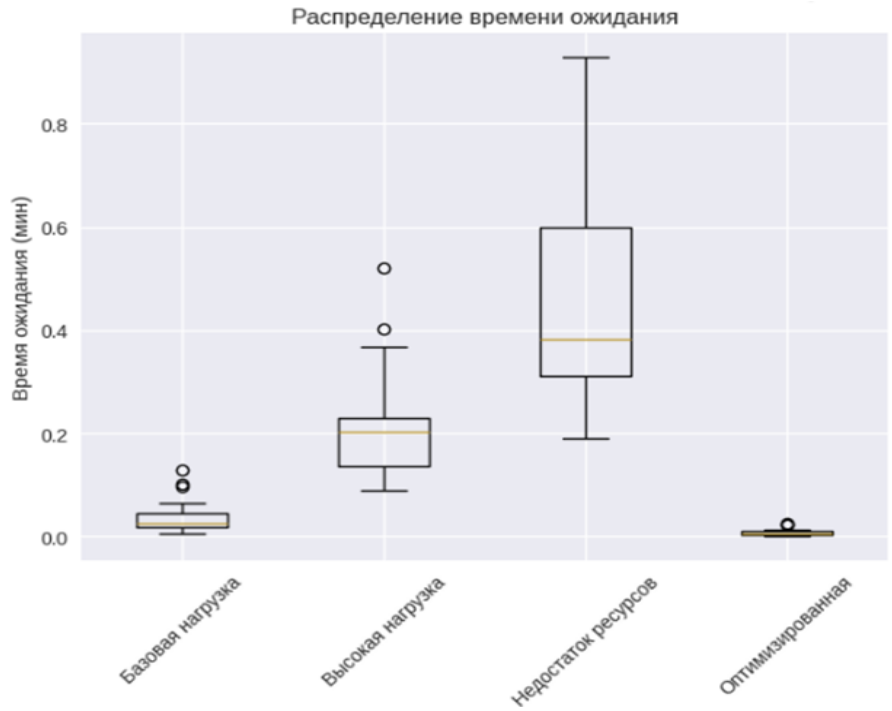

Рисунок 7 – Диаграмма размаха времени ожидания по сценариям
Источник: составлено авторами

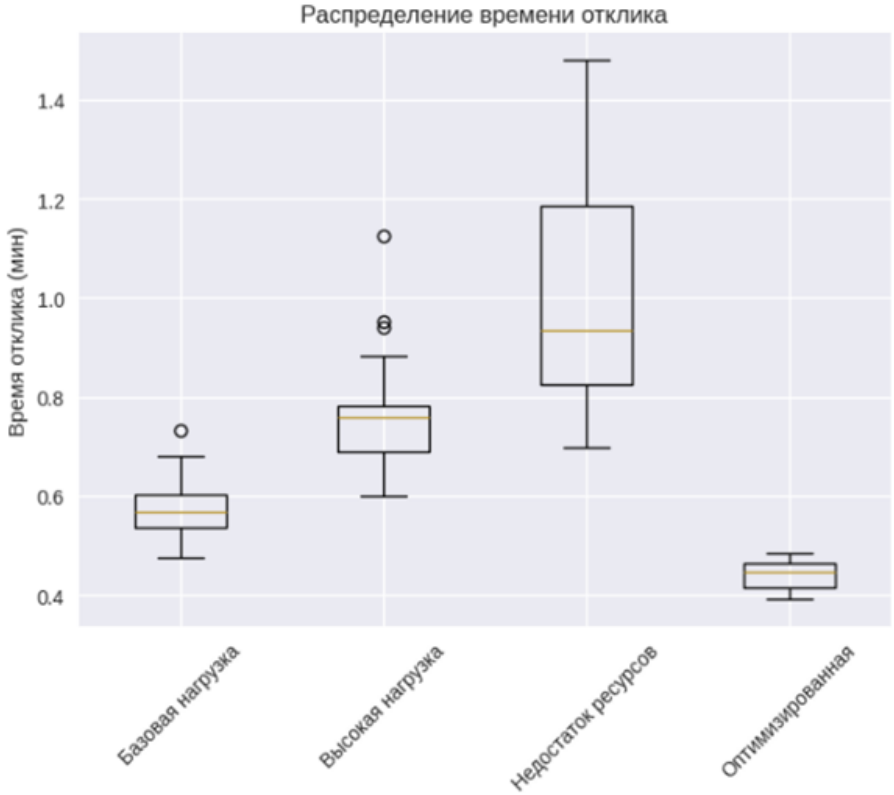

Рисунок 8 – Диаграмма размаха времени отклика по сценариям
Источник: составлено авторами

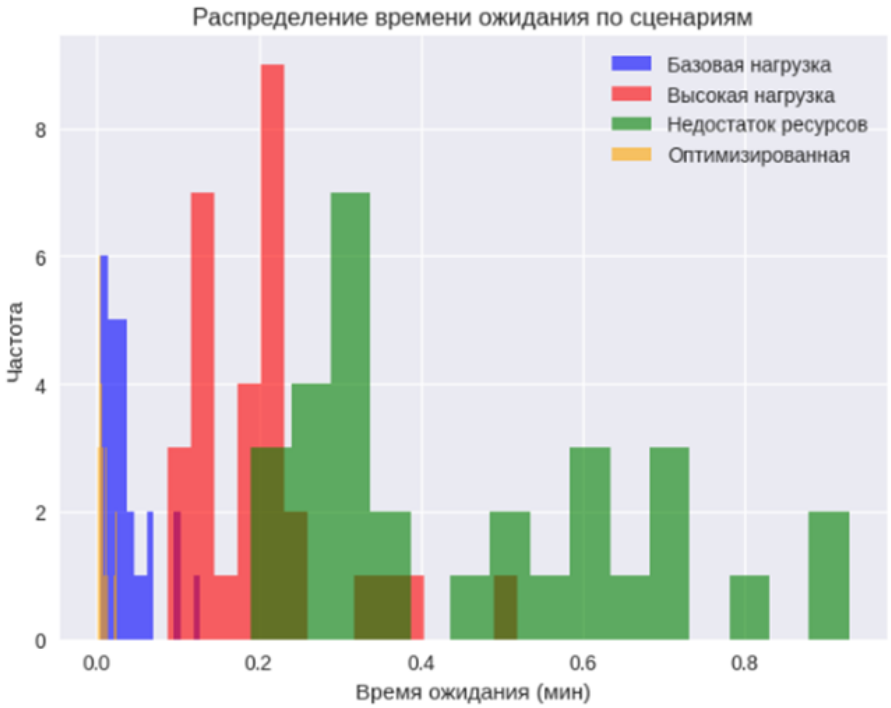

Рисунок 9 – Гистограмма распределения времени ожидания
Источник: составлено авторами

Таблица 7

Сводные показатели производительности по результатам имитационного моделирования
Источник: составлено авторами

Сценарий	Среднее время ожидания, мин.	Среднее время отклика, мин.	Загрузка системы, %	Вероятность очереди, %
Базовая нагрузка	0,04 ± 0,03	0,57 ± 0,05	45,9 ± 3,7	10,4 ± 3,8
Высокая нагрузка	0,21 ± 0,09	0,76 ± 0,11	81,5 ± 4,2	34,6 ± 4,4
Недостаток ресурсов	0,46 ± 0,21	1,00 ± 0,23	94,6 ± 6,5	50,0 ± 7,6
Оптимизированная	0,01 ± 0,01	0,44 ± 0,03	47,9 ± 2,6	4,3 ± 2,2

Анализ данных позволил сделать следующие выводы:

1) базовая конфигурация демонстрирует стабильную работу с минимальным временем ожидания (0,04 ± 0,03 мин) и умеренной загрузкой ресурсов (~46%);

2) сценарий с высокой нагрузкой подтверждает устойчивость системы: несмотря на рост загрузки до 81,5%, время отклика остается на приемлемом уровне (0,76 ± 0,11 мин);

3) сценарий с недостатком ресурсов выявляет «узкое место»: при сокращении числа серверов до двух и сохранении нагрузки загрузка достигает критических 94,6%, а время ожидания увеличивается в 10 раз по сравнению с базовым сценарием;

4) оптимизированная конфигурация (увеличение числа серверов и оптимизация процессов) показывает наилучшие результаты по времени отклика (0,44 ± 0,03 мин) при сохранении невысокой загрузки (~48%).

Выводы. В результате исследования разработан обоснованный комплекс требований и архитектурных принципов к интегрированной панели управления задачами машинного обучения. Проведённое анкетирование специалистов позволило выявить ключевые проблемы текущих MLOps-практик – фрагментированность инструментов, отсутствие единого интерфейса и сложность мониторинга процессов. На основе полученных данных определены приоритетные функциональные области будущей системы: централизованное управление задачами, интеграция с MLflow и Grafana, визуализация метрик и логов, система уведомлений и персонализированный интерфейс.

Сравнительный анализ пяти существующих платформ с применением метода интегральной оценки показал, что наибольшую зрелость демонстрируют коммерческие решения (Google Vertex AI, Amazon SageMaker), однако они ограничены по гибкости, тогда как открытые системы (Kubeflow) требуют значительных усилий по внедрению и поддержке. Это подтверждает необходимость разработки унифицированного решения, сочетающего гибкость open-source-инструментов и удобство коммерческих платформ.

Проведённое имитационное моделирование подтвердило эффективность предложенной архитектуры: оптимизированная конфигурация системы обеспечивает среднее время отклика менее 0,5 минуты при загрузке менее 50%, что свидетельствует о её устойчивости и высокой производительности.

Практическим результатом исследования является создание научно обоснованной основы для проектирования прототипа панели управления, обеспечивающей унификацию MLOps-процессов и снижение операционных затрат. Разработанные требования и архитектурные принципы направлены на решение проблем, характерных для компаний с распределённой инфраструктурой для МО и кросс-функциональными командами специалистов, где отсутствие единой точки управления затрудняет эксплуатацию моделей и взаимодействие между подразделениями.

В дальнейшем планируется реализация прототипа и его валидация в производственной среде компании-заказчика, соответствующей указанному организационному контексту.

Список источников

1. Fortune Business Insights. Machine Learning Market Size, Share & COVID-19 Impact Analysis, 2023–2030 [Электронный ресурс] – URL: https://www.fortunebusinessinsights.com/machine-learning-market-102226 (дата обращения: 09.11.2025). – Текст: электронный. (In Eng.).
2. Kreuzberger D., Kühl N., Hirschl S. MLOps: Overview, Definition, and Architecture // arXiv preprint arXiv:2210.11831, 2022. – DOI: 10.48550/arXiv.2205.02302. – Текст: электронный. (In Eng.).
3. Chapman P., Clinton J., Kerber R. et al. CRISP-DM 1.0: Step-by-step data mining guide. – SPSS Inc., 2000. (In Eng.).
4. CRISP-DM: проверенная методология для Data Scientist-ов // Хабр [Электронный ресурс]. – URL: https://habr.com/ru/companies/lanit/articles/328858/ (дата обращения: 09.11.2025). – Текст: электронный.
5. Polyzotis N., Roy S., Whang S.E., Zinkevich, M. Data Lifecycle Challenges in Production Machine Learning: A Survey // ACM SIGMOD Record. – 2018. – Vol. 47. – No. 2. – P. 17–28. (In Eng.).
6. Пантюхин М. А. Роль API в интеграции современных веб-сервисов и приложений // Научный аспект. – 2024. – № 5. – С. 1138.

References

1. Fortune Business Insights. Machine Learning Market Size, Share & COVID-19 Impact Analysis, 2023–2030. Available at: https://www.fortunebusinessinsights.com/machine-learning-market-102226. (Accessed: 09.11.2025).

2. Kreuzberger D., Kühl N., Hirschl S. MLOps: Overview, Definition, and Architecture. 2022. Available at: arXiv preprint arXiv:2210.11831.

3. Chapman P., Clinton J., Kerber R. et al. CRISP-DM 1.0: Step-by-step data mining guide. – SPSS Inc., 2000.
4. CRISP-DM: A Proven Methodology for Data Scientists. *Habr*. Available at: https://habr.com/ru/companies/lanit/articles/328858/. (Accessed: 09.11.2025). (In Russ.).

5. Polyzoti, N., Roy S., Whang S. E., Zinkevich, M. Data Lifecycle Challenges in Production Machine Learning: A Survey. *ACM SIGMOD Record*. 2018. Vol. 47. No. 2. pp. 17–28.
6. Pantyukhin M. A. The Role of API in the Integration of Modern Web Services and Applications. *Nauchniy aspekt*. 2024. No. 5. P. 1138. (In Russ.).

7. Репетий Е. О., Хохлова Е. И. Роль RESTful API в интеграции современных веб-сервисов // Научный аспект. – 2024. – Т. 44. – № 4. – С. 5793–5798.
8. Шранк А. А. Методы и средства обработки данных // Наука, общество, производство и промышленность: актуальные проблемы и перспективы. – – 2023. – С. 32–34.
9. Singla A. Machine Learning Operations (MLOps): Challenges and Strategies // Journal of Knowledge Learning and Science Technology. – 2023. – Vol. 2. – No. 3. – P. 337–341. (In Eng.).
10. Ангапов В. Д. Обзор современных облачных платформ для целей машинного обучения // Проблемы науки. – 2023. – № 7 (185) [Электронный ресурс]. – URL: https://cyberleninka.ru/article/n/obzor-sovremennyh-oblachnyh-platform-dlya-tseley-mashinnogo-obucheniya. (дата обращения: 09.11.2025). – Текст: электронный.
11. Mayring P. Qualitative Content Analysis. // Forum Qualitative Sozialforschung. 2000 [Электронный ресурс]. – URL: https://www.researchgate.net/publication/215666096_Qualitative_Content_Analysis (дата обращения: 18.11.2025). – Текст: электронный. (In Eng.).
12. Хорошилов Д. А., Мельникова О. Т. Метод тематического анализа в изучении представлений о женском лидерстве // Организационная психология. 2020. № 3 [Электронный ресурс]. – URL: https://cyberleninka.ru/article/n/metod-tematicheskogo-analiza-v-izuchenii-predstavleniy-o-zhenskom-liderstve (дата обращения: 18.11.2025). – Текст: электронный.
13. Попова В. Б. Выбор статистических критериев при анализе данных // Проблемы национальной экономики в цифрах статистики: Материалы IX международной научно-практической конференции, Тамбов, 13.12.2022. – Тамбов: Издательский дом «Державинский», 2023. – С. 439–446.
14. Кравченко Т., Брускин С. Приоритизация требований для эффективной поддержки процесса коммуникаций с клиентами коммерческого банка // Бизнес-информатика. – 2017. – Т. 11. – №. 2. – С. 7–16.
15. Кельтон В., Лоу А. Имитационное моделирование. Классика CS. 3-е изд. – СПб.: Питер, 2004. – 848 с.
16. Крысанов И. С. Имитационное моделирование на примере метода Монте-Карло симуляции // Фармакоэкономика. Современная фармакоэкономика и фармакоэпидемиология. 2008. № 2 [Электронный ресурс]. – URL: https://cyberleninka.ru/article/n/imitatsionnoe-modelirovanie-na-primere-metoda-monte-karlo-simulyatsii. (дата обращения: 09.11.2025). – Текст: электронный.
17. Горшков П. В., Бессмертный И. А., Клименков С. В. Методы и алгоритмы управления процессами разработки программных продуктов //

7. Repetiy E. O., Khokhlova E. I. The Role of RESTful API in the Integration of Modern Web Services. *Nauchniy aspekt*. 2024. Vol. 44. No. 4. pp. 5793–5798. (In Russ.).
8. Shrank A. A. Methods and Tools for Data Processing. *Nauka, obshchestvo, proizvodstvo i promysh-lennost': aktual'nyye problemy i perspektivy*. 2023. pp. 32–34. (In Russ.).
9. Singla A. Machine Learning Operations (MLOps): Challenges and Strategies. *Journal of Knowledge Learning and Science Technology*. 2023. Vol. 2. No. 3. pp. 337–341.
10. Angapov V. D. Review of Modern Cloud Platforms for Machine Learning. *Problemy nauki*. 2023. No. 7 (185). Available at: https://cyberleninka.ru/article/n/obzor-sovremennyh-oblachnyh-platform-dlya-tseley-mashinnogo-obucheniya. (Accessed: 09.11.2025). (In Russ.).

11. Mayring P. Qualitative Content Analysis. *Forum Qualitative Sozialforschung*. 2000. Available at: https://www.researchgate.net/publication/215666096_Qualitative_Content_Analysis (Accessed: 18.11.2025).

12. Khoroshilov D. A., Melnikova O. T. Thematic in the Study of Ideas about Women's Leadership. *Organizatsionnaya psikhologiya*. 2020. No. 3. Available at: https://cyberleninka.ru/article/n/metod-tematicheskogo-analiza-v-izuchenii-predstavleniy-o-zhenskom-liderstve (Accessed: 18.11.2025). (In Russ.).

13. Popova V. B. Selection of Statistical Criteria in Data Analysis. *Problems of the National Economy in Statistics: Proceedings of the 9th International Scientific and Practical Conference, Tambov, December 13, 2022*. Tambov: Derzhavinsky Publishing House, 2023. pp. 439–446. (In Russ.).
14. Kravchenko T., Bruskin S. Prioritization of Requirements for Effective Support of Communication Process with Clients of a Commercial Bank. *Business Informatics*. 2017. Vol. 11. No. 2. pp. 7–16. (In Russ.).
15. Kelton V., Lowe A. Simulation Modeling. CS Classics. 3rd ed. *St. Petersburg: Piter*. 2004. 848 p. (In Russ.).
16. Krysanov I. S. Simulation Modeling Using the Monte Carlo Simulation Method. *Farmakoekonomika. Sovremennaya farmako-ekonomika i farmakoepidemiologiya*. 2008. No. 2. Available at: https://cyberleninka.ru/article/n/imitatsionnoe-modelirovanie-na-primere-metoda-monte-karlo-simulyatsii. (Accessed: 09.11.2025). (In Russ.).

17. Gorshkov P. V., Bessmertny I. A., Klimenkov S. V. Methods and Algorithms for Managing Software Development Processes. *Ekonomika. Pravo.*

Экономика. Право. Инновации. – 2025. – Т. 13. – № 3. С. 59–68. – DOI: 10.17586/2713-1874-2025-3-59-68. – Текст: электронный.

18. Кирпичников А.П., Титовцев А. С., Фадхкал З. Некоторые особенности числовых характеристик многоканальных систем массового обслуживания открытого типа // Вестник Казанского технологического университета. – 2015. – № 2. – URL: https://cyberleninka.ru/article/n/nekotorye-osobennosti-chislovyh-harakteristik-mnogokanalnyh-sistem-massovogo-obsluzhivaniya-otkrytogo-tipa. (дата обращения: 09.11.2025). – Текст: электронный.

Innovacii. 2025. Vol. 13. No. 3. pp. 59–68. (In Russ.). DOI: 10.17586/2713-1874-2025-3-59-68.

18. Kirpichnikov A. P., Titovtsev A. S., Fadhkal Z. Some Features of Numerical Characteristics of Open-Type Multi-Channel Queuing Systems. *Vestnik Kazanskogo tekhnologicheskogo universiteta*. 2015. No. 2. Available at: https://cyberleninka.ru/article/n/nekotorye-osobennosti-chislovyh-harakteristik-mnogokanalnyh-sistem-massovogo-obsluzhivaniya-otkrytogo-tipa. (Accessed: 09.11.2025). (In Russ.).

Статья поступила в редакцию 17.11.2025; одобрена после рецензирования 10.12.2025; принята к публикации 12.12.2025.
The article was submitted 17.11.2025; approved after reviewing 10.12.2025; accepted for publication 12.12.2025.

Вклад авторов в исследование:

Н. В. Добренко – научное редактирование текста, формирование идеи, формулирование исследовательских целей и задач, разработка методологии исследования, управление исследовательским процессом

В. Ю. Бабан – литературный обзор, проведение опроса, определение архитектурных принципов, воспроизводство результатов, подготовка рукописи статьи

А. В. Чернышева – моделирование, редактирование выводов

А. И. Говоров – редактирование текста, анализ данных