

МОЛОДОЙ УЧЁНЫЙ

ISSN 2072-0297

МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ



29 2026
ЧАСТЬ I

16+

Молодой ученый

Международный научный журнал

№ 29 (632) / 2026

Издается с декабря 2008 г.

Выходит еженедельно

Главный редактор: Ахметов Ильдар Геннадьевич, кандидат технических наук

Редакционная коллегия:

Жураев Хусниддин Олтинбоевич, доктор педагогических наук (Узбекистан)
Иванова Юлия Валентиновна, доктор философских наук
Каленский Александр Васильевич, доктор физико-математических наук
Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)
Куташов Вячеслав Анатольевич, доктор медицинских наук
Лактионов Константин Станиславович, доктор биологических наук
Сараева Надежда Михайловна, доктор психологических наук
Абдрасилов Турганбай Курманбаевич, доктор философии (PhD) по философским наукам (Казахстан)
Авдеюк Оксана Алексеевна, кандидат технических наук
Айдаров Оразхан Турсункожаевич, кандидат географических наук (Казахстан)
Алиева Тарана Ибрагим кызы, кандидат химических наук (Азербайджан)
Ахметова Валерия Валерьевна, кандидат медицинских наук
Бердиев Эргаш Абдуллаевич, кандидат медицинских наук (Узбекистан)
Брезгин Вячеслав Сергеевич, кандидат экономических наук
Данилов Олег Евгеньевич, кандидат педагогических наук
Дёмин Александр Викторович, кандидат биологических наук
Дядюн Кристина Владимировна, кандидат юридических наук
Желнова Кристина Владимировна, кандидат экономических наук
Жуйкова Тамара Павловна, кандидат педагогических наук
Игнатова Мария Александровна, кандидат искусствоведения
Искаков Руслан Маратбекович, кандидат технических наук (Казахстан)
Калдыбай Кайнар Калдыбайулы, доктор философии (PhD) по философским наукам (Казахстан)
Кенесов Асхат Алмасович, кандидат политических наук
Коварда Владимир Васильевич, кандидат физико-математических наук
Комогорцев Максим Геннадьевич, кандидат технических наук
Котляров Алексей Васильевич, кандидат геолого-минералогических наук
Кузьмина Виолетта Михайловна, кандидат исторических наук, кандидат психологических наук
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)
Кучерявенко Светлана Алексеевна, кандидат экономических наук
Лескова Екатерина Викторовна, кандидат физико-математических наук
Макеева Ирина Александровна, кандидат педагогических наук
Матвиенко Евгений Владимирович, кандидат биологических наук
Матроскина Татьяна Викторовна, кандидат экономических наук
Матусевич Марина Степановна, кандидат педагогических наук
Мусаева Ума Алиевна, кандидат технических наук
Насимов Мурат Орленбаевич, кандидат политических наук (Казахстан)
Паридинова Ботагоз Жаппаровна, магистр философии (Казахстан)
Прончев Геннадий Борисович, кандидат физико-математических наук
Рахмонов Азизхон Боситхонович, доктор педагогических наук (Узбекистан)
Семахин Андрей Михайлович, кандидат технических наук
Сенцов Аркадий Эдуардович, кандидат политических наук
Сенюшкин Николай Сергеевич, кандидат технических наук
Султанова Дилшода Намозовна, доктор архитектурных наук (Узбекистан)
Титова Елена Ивановна, кандидат педагогических наук
Ткаченко Ирина Георгиевна, кандидат филологических наук
Федорова Мария Сергеевна, кандидат архитектуры
Фозилов Садриддин Файзуллаевич, кандидат химических наук (Узбекистан)
Яхина Асия Сергеевна, кандидат технических наук
Ячинова Светлана Николаевна, кандидат педагогических наук

Международный редакционный совет:

Айрян Заруи Геворковна, кандидат филологических наук, доцент (Армения)
Арошидзе Паата Леонидович, доктор экономических наук, ассоциированный профессор (Грузия)
Атаев Загир Вагитович, кандидат географических наук, профессор (Россия)
Ахмеденов Кажмурат Максutowич, кандидат географических наук, ассоциированный профессор (Казахстан)
Бидова Бэла Бертовна, доктор юридических наук, доцент (Россия)
Борисов Вячеслав Викторович, доктор педагогических наук, профессор (Украина)
Буриев Хасан Чутбаевич, доктор биологических наук, профессор (Узбекистан)
Велковска Гена Цветкова, доктор экономических наук, доцент (Болгария)
Гайич Тамара, доктор экономических наук (Сербия)
Данатаров Агахан, кандидат технических наук (Туркменистан)
Данилов Александр Максимович, доктор технических наук, профессор (Россия)
Демидов Алексей Александрович, доктор медицинских наук, профессор (Россия)
Досманбетов Динар Бакбергенович, доктор философии (PhD), проректор по развитию и экономическим вопросам (Казахстан)
Ешиев Абдыракман Молдоалиевич, доктор медицинских наук, доцент, зав. отделением (Кыргызстан)
Жолдошев Сапарбай Тезекбаевич, доктор медицинских наук, профессор (Кыргызстан)
Игисинов Нурбек Сагинбекович, доктор медицинских наук, профессор (Казахстан)
Кадыров Кутлуг-Бек Бекмурадович, доктор педагогических наук, и.о. профессора, декан (Узбекистан)
Каленский Александр Васильевич, доктор физико-математических наук, профессор (Россия)
Козырева Ольга Анатольевна, кандидат педагогических наук, доцент (Россия)
Колпак Евгений Петрович, доктор физико-математических наук, профессор (Россия)
Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)
Куташов Вячеслав Анатольевич, доктор медицинских наук, профессор (Россия)
Кыят Эмине Лейла, доктор экономических наук (Турция)
Лю Цзюань, доктор филологических наук, профессор (Китай)
Малес Людмила Владимировна, доктор социологических наук, доцент (Украина)
Нагервадзе Марина Алиевна, доктор биологических наук, профессор (Грузия)
Нурмамедли Фазиль Алигусейн оглы, кандидат геолого-минералогических наук (Азербайджан)
Прокопьев Николай Яковлевич, доктор медицинских наук, профессор (Россия)
Прокофьева Марина Анатольевна, кандидат педагогических наук, доцент (Казахстан)
Рахматуллин Рафаэль Юсупович, доктор философских наук, профессор (Россия)
Ребезов Максим Борисович, доктор сельскохозяйственных наук, профессор (Россия)
Сорока Юлия Георгиевна, доктор социологических наук, доцент (Украина)
Султанова Дилшода Намозовна, доктор архитектурных наук (Узбекистан)
Узаков Гулом Норбоевич, доктор технических наук, доцент (Узбекистан)
Федорова Мария Сергеевна, кандидат архитектуры (Россия)
Хоналиев Назарали Хоналиевич, доктор экономических наук, старший научный сотрудник (Таджикистан)
Хоссейни Амир, доктор филологических наук (Иран)
Шарипов Аскар Калиевич, доктор экономических наук, доцент (Казахстан)
Шуклина Зинаида Николаевна, доктор экономических наук (Россия)

На обложке изображен *Бьёрн Страуструп* (1950), датский программист.

Бьёрн Страуструп родился в датском городе Орхус. После окончания школы он поступил в местный университет на отделение компьютерных технологий. На его решение повлиял школьный учитель математики, который привил мальчику любовь к этой науке.

Юноша впервые познакомился с компьютером уже в стенах вуза. Algol-60 был первым языком программирования, который выучил Бьёрн.

Орхусский университет входил в топ-100 высших учебных заведений мира. Страуструп со страхом думал о том, что ему придется работать учителем математики, если не найдется работа в сфере программирования. Он был убежден, что математика должна иметь практическое приложение.

Однако на втором курсе он понял, что программирование и компьютерные технологии в целом — это не только и не столько математика.

В университетские годы Страуструп подрабатывал тем, что писал небольшие программы для подразделения компании Burroughs в Орхусе. Тогда он понял, что коммерческая разработка имеет мало общего с написанием программ для себя или друзей и тем более с выполнением лабораторных работ в вузе. Кроме того, студент разрабатывал ПО и для нужд университета. В 1975 году Страуструп окончил его и получил степень магистра.

Затем молодой человек переехал в Англию, чтобы продолжить свое образование в Кембриджском университете. Там в вычислительной лаборатории он занимался проектированием распределенных систем. В 1979 году Бьёрн получил докторскую степень. Тогда же Страуструп вместе с женой Мэриэн переехал в Нью-Джерси (США), где стал работать в центре компьютерных исследований фирмы Bell Labs.

Когда Страуструп занимался исследованиями в фирме, ему потребовалось написать несколько имитационных программ для моделирования распределенных вычислений. SIMULA-67 — объектно-ориентированный язык — мог бы стать идеальным инструментом для решения подобных задач, если бы не его сравнительно низкая скорость выполнения программ. До этого в сферу интересов Страуструпа не входила разработка языков программирования. Его интересовало только микропрограммирование, операционные системы, архитектура вычислительных машин, разработка виртуальных машин. Правда, еще до окончания университета он успел изучить двадцать языков программирования. Но, несмотря на это, Бьёрн утверждает, что большого интереса к языкам он все-таки не испытывал. Тем не менее, Страуструп принял решение начать работу над новым языком.

Так был создан язык программирования C++, первоначально получивший название «Си с классами» (C with

classes). Название C++ придумал Рик Мэсцитти. ++ — это оператор инкремента в C, который как бы намекает на то, что язык C++ — это нечто большее, чем просто C. В 1983 году язык подвергся значительным изменениям. Страуструп разработал его таким образом, чтобы код не компилировался в машинный язык, а преобразовывался в набор команд препроцессора языка C. Это открывало к нему доступ сотням тысяч C-программистов, имевших соответствующий компилятор.

В 1985 году Страуструп опубликовал одну из самых широко известных книг «Язык программирования C++», которая выдержала четыре издания (1985, 1991, 1997, 2000) и была переведена на 19 языков.

В 1990 году вышла очередная книга Страуструпа — «Справочное руководство по языку программирования C++ с комментариями» (The Annotated C++ Reference Manual), которая в дальнейшем была удостоена награды за непревзойденное мастерство в области технической документации, по мнению журнала Dr. Dobbs' Journal. Кроме того, по версии журнала Fortune magazine, Страуструп был признан одним из «двенадцати лучших молодых американских ученых».

А Страуструп тем временем продолжал работать в AT&T Bell Labs, где возглавлял подразделение широко-масштабных программных исследований, активно занимался совершенствованием своего языка и созданием его стандарта. Стандарт ANSI/ISO C++ был выпущен в 1999 году. Страуструп был главой отдела исследований программирования в Bell Labs с 1995 по 2002 год.

С 2002 года Бьёрн является профессором Техасского университета A&M (Texas A&M University). Кроме того, он сотрудничает с Колумбийским университетом как приглашенный профессор в области компьютерных наук.

В 2014 году Бьёрну предложил работу один из крупнейших финансовых конгломератов мира — Morgan Stanley. Страуструп начал сотрудничать с этой фирмой, так как решил вернуться из сферы чистой педагогики к решению реальных проблем.

Бьёрн Страуструп был избран членом Национальной академии инженерии в США в 2004 году с формулировкой «за создание языка C++». Как первый компьютерный ученый, он в 2005 году получил премию Уильяма Проктера за научные достижения американского научно-исследовательского сообщества. Он также получил награду компьютерного сообщества IEEE «за начало разработки и коммерциализации объектно-ориентированной технологии создания программ и за большие изменения, внесенные в бизнес и индустрию».

*Информацию собрала ответственный редактор
Екатерина Осянина*

СОДЕРЖАНИЕ

ХИМИЯ

Радул К. А.

Влияние легирования оксидом тербия и совместного введения оксидов самария и кобальта на параметры железо-иттриевого граната для СВЧ-фазовращателей 1

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Васюрин А. В.

Сценарное автоматизированное тестирование аутентификации веб-приложения с помощью Selenium WebDriver 4

Гаас Д. А., Гаас С. А.

Влияние качества фильтрации воздуха на отказоустойчивость серверного оборудования и срок его службы 6

Галкин А. А.

Эволюция систем предотвращения вторжений: от сигнатурного анализа к поведенческим методам и машинному обучению 9

Дунаев Л. Д., Хусяинова А. М., Щавровская П. К.

Сравнительный анализ методов защиты веб-приложений от SQL-инъекций и XSS-атак 12

Жарков Н. А.

Микрофронтенд-архитектура: понятие и предпосылки возникновения 16

Жарков Н. А.

Сравнительный анализ и ограничения микрофронтенд-архитектуры 18

Жарков Н. А.

Динамическая интеграция микрофронтендов: подходы и инструменты 20

Зарембо И. А., Мительман Ю. Е.

Разработка интерактивной виртуальной лаборатории для исследования основной и высшей волны в круглом волноводе 21

Козырев П. М.

Влияние UX-дизайна на эффективность работы пользователей в информационных системах 26

Козырев П. М.

Влияние кэширования на производительность веб-приложений 28

Козырев П. М.

Влияние микросервисной архитектуры на сопровождаемость программных систем 30

Козырев П. М.

Использование графовых баз данных для анализа связанной информации 32

Козырев П. М.

Использование компьютерного зрения для распознавания объектов в реальном времени ... 34

Лихинин А. Е.

Прогностические модели на основе машинного обучения как инструмент оптимизации закупочной деятельности в условиях волатильности сырьевых рынков 37

Пронин К. Н., Леонтьева Д. С.

Интеграция Process Mining и искусственного интеллекта для предиктивного управления бизнес-процессами 39

Селезнёв А. И., Селезнёв И. Л.

Особенности использования нейронных сетей в управлении беспилотным автомобилем 45

ТЕХНИЧЕСКИЕ НАУКИ

Шутов В. М.

Системы фильтрации топлива в дизельных двигателях сельскохозяйственной техники 50

МЕДИЦИНА

Ильясов А. С., Серик М. С., Иманбек Б. К.

Оценка организации и качества амбулаторной медицинской помощи в первичном звене здравоохранения: взгляд медицинских работников 52

Чкалин И. В.

Нестероидные противовоспалительные препараты в хирургии: баланс между анальгезией и риском периоперационных осложнений 56

ХИМИЯ

Влияние легирования оксидом тербия и совместного введения оксидов самария и кобальта на параметры железо-иттриевого граната для СВЧ-фазовращателей

Радул Ксения Александровна, студент

Санкт-Петербургский государственный технологический институт (технический университет)

В статье исследуется влияние оксида тербия и совместного введения оксидов самария и кобальта на электромагнитные свойства ферритовых материалов на основе железо-иттриевого граната, предназначенных для СВЧ-фазовращателей, работающих на высоком уровне мощности. Основное внимание уделено ширине линии резонанса спиновых волн (ΔH_k).

Ключевые слова: железо-иттриевый гранат, оксиды редкоземельных металлов, СВЧ-феррит.

Феррит, относящийся к группе гранатов, имеющих малые потери и высокую термостабильность намагниченности насыщения для применения в невзаимных приборах среднего уровня мощности, используется для производства элементов фазовращателей, которые работают в режимах, далеких от ферромагнитного резонанса, не поглощая энергию СВЧ-поля, а изменяя фазовую скорость волн.

В современном мире фазовращатели незаменимы в следующих отраслях промышленности:

- 1) спутниковая связь;
- 2) радары и военные технологии;
- 3) измерительное оборудование;
- 4) энергетика (электросети);
- 5) оптические сети и другие сферы.

Применение в этих отраслях требует малых потерь при работе приборов на высоких мощностях. Этому условию не соответствовал исходный состав феррограната, что приводило к низкому выходу годных пластин. Возникла необходимость модифицировать материал путем введения легирующих добавок, способных изменить его характеристики.

Совместное введение оксидов самария (Sm_2O_3) и кобальта (Co_3O_4) в состав феррита, имеющего структуру граната, позволяет целенаправленно изменять их магнитные параметры. Легирование осуществлялось на этапе приготовления шихты путем смешения порошкообразных оксидов с последующей реализацией стандартного технологического процесса, включающего помол, ферритизацию, прессование и обжиг.

Как показано в литературе, ионы Sm^{3+} , обладающие ионным радиусом 0,96 Å, замещают ионы иттрия (Y^{3+} , 0,89 Å) в додекаэдрических позициях кристаллической

решетки граната, что приводит к увеличению параметра решетки и влияет на обменные взаимодействия в подрешетках феррита [1]. Ионы $\text{Co}^{2+}/\text{Co}^{3+}$ при этом занимают преимущественно октаэдрические позиции, создавая дополнительный вклад в магнитоупругую энергию и изменяя релаксационные характеристики материала [2].

Совместное легирование редкоземельными элементами и кобальтом может приводить к существенному изменению структурных и магнитных характеристик. В частности, в системах $\text{Co}_{1+x}\text{Sm}_x\text{Fe}_{2-2x}\text{O}_4$ наблюдается увеличение размера кристаллитов и изменение коэрцитивной силы [3].

Согласно результатам, полученным в рамках работы (таблица 1), совместное введение Co_3O_4 и Sm_2O_3 в железо-иттриевый гранат дает промежуточные результаты между легированием только самарием или только кобальтом, сужая температурный диапазон получения годных изделий. Данное явление связано с конкуренцией эффектов от двух типов легирующих добавок: с одной стороны, ионы самария увеличивают ширину линии спиновых волн, подавляя нелинейные процессы [3], с другой — ионы кобальта создают дополнительные каналы магнитной релаксации и влияют на магнитоstriction [3].

Ширина линии резонанса ΔH_k составляла 16,9–18,9 Э, что позволяет избежать больших потерь при работе фазовращателя на высоком уровне мощности. Удовлетворительный выход годных (выше 60 %) фиксировался только в узких интервалах разброса плотности партии до 0,008 г/см³. А 100 % выхода годных изделий удалось достичь при диапазоне плотности в партии всего 0,001 г/см³, что может привести к затруднению воспроизводимости результатов и необходимостью переобжига.

Таблица 1. Параметры аттестации партий, легированных оксидами самария и кобальта

Номер партии	Плотность, г/см ³	4 пMs, Гс	ΔH _k , Э	Выход годных, %
1	5,679–5,685	885	17,4	0
2	5,637–5,651	876	17,8	40
3	5,670–5,672	869	16,9	88
4	5,677–5,684	899	17,1	0
5	5,671–5,679	905	17,9	71
6	5,647–5,656	890	17,7	0
7	5,666–5,667	860	17,6	100

Дополнительно было рассмотрено влияния ведения оксида тербия (Tb_4O_7) в состав шихты приводит к формированию твердого раствора замещения $Tb_xY_{3-x}Fe_5O_{12}$, в котором ионы тербия занимают додекаэдрические позиции кристаллической решетки граната. Ключевая особенность тербийсодержащих ферритов гранатов заключается в сильном влиянии тербия на анизотропию и магнито-стрикционные свойства материала.

Принципиальное значение для применения в фазовращателях высокого уровня мощности имеет влияние тербия на параметр ΔH_k — ширину линии спиновых волн, определяющую нелинейные потери при больших СВЧ-мощностях.

Ионы тербия, обладающие незамороженным орбитальным моментом, создают дополнительные каналы релаксации для спиновых волн, что способствует повы-

шению порога возникновения нелинейных процессов. В технических требованиях к ферритовым материалам для фазовращателей X-диапазона ширина линии спиновых волн ΔH_k должна составлять не менее 0,8 кА/м (что соответствует примерно 10 Э), и введение оптимальных концентраций тербия позволяет достичь этого значения [4].

Анализ параметров аттестации партий, легированных оксидом тербия, представленных в таблице 2, показал, что введение оксида тербия позволило стабильно получать высокий выход годных изделий от 60 до 100 %. Плотность образцов с тербием находится в диапазоне 5,623–5,680 г/см³. Ширина линии резонанса спиновых волн ΔH_k у легированных оксидом тербия образцов составила 16,9–18,1 Э, что соответствует значениям полученного феррита, легированного оксидом самария. Таким образом, по уровню потерь влияние тербия не уступает влиянию самария.

Таблица 2. Параметры аттестации партий, легированных оксидом тербия

Номер партии	Плотность, г/см ³	4 пMs, Гс	ΔH _k , Э	Выход годных, %
8	5,623–5,633	843	16,9	90
9	5,635–5,648	852	17,1	60
10	5,648–5,651	865	18,1	90
11	5,650–5,658	862	18,0	90
12	5,679–5,680	865	17,9	100

Стоит отметить, что образцы, легированные оксидом тербия, нуждаются в более высоких температурах обжига, нежели феррит, легированный оксидами самария и кобальта, так как оксид тербия является более тугоплавкой добавкой.

Проведенное исследование показало, что как совместное легирование железо-иттриевого граната оксидами самария и кобальта, так и модифицирование оксидом тербия позволяют достичь достаточных значений ширины линии спиновых волн, необходимых для работы СВЧ-фазовращателей на высоком уровне мощности без больших потерь.

Введение оксида тербия обеспечивает стабильно высокий выход годных изделий (от 60 до 100 %) в широком интервале плотностей партий, тогда как при совместном легировании Sm_2O_3 и Co_3O_4 удовлетворительный выход достигается лишь в узких интервалах плотности, а 100 % выход годных изделий получен только при диапазоне плотности в партии 0,001 г/см³, что существенно услож-

няет воспроизводимость технологии и требует дополнительных переобжигов.

Кроме того, тербий, обладая незамороженным орбитальным моментом, эффективно подавляет нелинейные потери за счет создания дополнительных релаксационных каналов, не уступая по этому показателю самарию. При этом его использование исключает конкурирующее влияние разнородных добавок, характерное для системы Sm_2O_3 и Co_3O_4 , и позволяет упростить контроль технологического процесса.

Несмотря на необходимость более высоких температур обжига, обусловленных большей тугоплавкостью оксида тербия, данный недостаток компенсируется высокой повторяемостью результатов и снижением доли брака. Таким образом, оксид тербия является более перспективной легирующей добавкой для производства ферритовых материалов на основе железо-иттриевого граната, предназначенных для мощных СВЧ-фазовращателей.

Литература:

1. Cunningham, J.R., Jr. Samarium Substitutions in Yttrium Iron Garnet / J. R. Cunningham, Jr., E. E. Anderson — DOI: 10.1063/1.1984598 // Journal of Applied Physics. — 1960. — Vol. 31, No. 5. — P. S45–S46.
2. Григорьева, Н. Б. Ферритовый материал: пат. 2776991 Рос. Федерация, МПК H01F 1/34 (2006.01), C04B 35/26 (2006.01) / Н. Б. Григорьева, Н. А. Ковалёва, Т. А. Лисейцева [и др.]; заявитель и патентообладатель Акционерное общество «Научно-исследовательский институт «Феррит-Домен». — № 2022104202; заявл. 18.02.2022; опубл. 01.09.2022, Бюл. № 25. — 6 с.: 1 табл.
3. Асао, Х. Фазовращатель с двойным режимом, нечувствительный к магнитострикции / Х. Асао, М. Мацунага, Ф. Такеда — DOI: 10.1002/ecja.4410691110 // Electronics and Communications in Japan (Part I: Communications). — 1986. — Т. 69, № 11. — С. 94–100.
4. Hansen, P. Anisotropy and magnetostriction of cobalt-substituted yttrium iron garnet / P. Hansen, W. Tolksdorf, R. Krishnan // Physical Review B. — 1977. — Vol. 16, No. 9. — P. 3973–3980.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Сценарное автоматизированное тестирование аутентификации веб-приложения с помощью Selenium WebDriver

Васюрин Артём Викторович, студент
Московский политехнический университет

В статье предложен сценарный подход к автоматизированному тестированию аутентификации веб-приложения. Тестовый случай описывается предусловиями, входными данными, действиями, условиями ожидания и ожидаемым состоянием интерфейса. Подход реализован на Java 17 с применением Selenium WebDriver 4.23.0 и JUnit 5; сформирована матрица положительных, негативных и граничных сценариев. В учебном проекте реализован воспроизводимый положительный UI-тест как основа для расширения набора проверок.

Ключевые слова: автоматизированное тестирование, Selenium WebDriver, JUnit 5, UI-тестирование, аутентификация, явное ожидание, тестовый сценарий.

Введение

Аутентификация относится к критичным пользовательским сценариям веб-приложения: до её успешного выполнения пользователь не получает доступ к защищённым функциям. Проверка такого поведения относится к функциональному тестированию пользовательского интерфейса [5, 6]. При ручной проверке тестировщик многократно открывает форму, вводит различные данные, отправляет запрос и сопоставляет реакцию интерфейса с ожидаемым результатом. Автоматизация этого цикла уменьшает объём повторяемой работы и делает проверку пригодной для регулярного запуска после изменений приложения.

Основная техническая проблема UI-тестов связана с асинхронностью браузерного интерфейса. Элемент может присутствовать в DOM, но оставаться невидимым или недоступным для взаимодействия. Selenium предупреждает о непредсказуемости совместного применения неявных и явных ожиданий и рекомендует не смешивать их в одном сценарии [1]. Поэтому надёжность теста определяется не только последовательностью команд, но и явно заданными условиями готовности интерфейса.

Цель работы — разработать воспроизводимую сценарную методику проверки аутентификации, фиксирующую предусловия, тестовые данные, действия, условия ожидания и наблюдаемый результат. Для достижения цели формализован тестовый случай, определена последовательность выполнения, разработана матрица сценариев и реализован положительный UI-тест.

Материалы и методы

Технологическую основу составляют Java 17, Selenium WebDriver 4.23.0 и JUnit 5. WebDriver предоставляет единый интерфейс управления браузером, а JUnit организует жизненный цикл тестов и выполнение утверждений [2]. Требования к протоколу и модели управления браузером определены спецификацией WebDriver [4]. Для синхронизации используются явные ожидания, которые приостанавливают выполнение до наступления конкретного состояния элемента или страницы [1].

Тестовый случай представляется кортежем $T = (P, D, A, W, E)$, где P — предусловия; D — тестовые данные; A — последовательность действий; W — условия ожидания; E — ожидаемое наблюдаемое состояние. Добавление W в модель принципиально важно для UI-теста: после действия браузер может не сразу перейти в состояние, пригодное для проверки.



JUnit 5 • подготовка • действие • проверка • отчёт об ошибке

Рис. 1. Последовательность выполнения UI-теста аутентификации

Для поиска элементов выбираются локаторы, связанные с назначением элемента, а не с его положением в разметке. Предпочтение отдаётся стабильным идентификаторам, атрибутам name, data-testid или семантическим CSS-селекторам. После отправки формы применяется ожидание признака завершения перехода: появления элемента целевой страницы, изменения URL, исчезновения формы или отображения сообщения об ошибке.

Каждый тест должен начинаться с известного состояния. Подготовка браузера и открытие страницы выполняются до сценария, а очистка ресурсов — после него. Тестовые данные не должны зависеть от результата предыдущего запуска. Такая организация делает сбой воспроизводимым и позволяет определить этап, на котором ожидаемое состояние не было достигнуто.

Таблица 1. Сценарии проверки аутентификации

Сценарий	Тестовые данные	Проверяемое состояние
Положительная аутентификация	Допустимые логин и пароль	Открыта защищённая страница или отображён её устойчивый признак
Неверный пароль	Корректный логин, неверный пароль	Показано сообщение об отказе; доступ не предоставлен
Неизвестный пользователь	Логин отсутствует в системе	Показано нейтральное сообщение об ошибке
Пустые поля	Пустой логин и/или пароль	Сработала клиентская или серверная валидация
Граничная длина	Минимальные и максимальные допустимые значения	Интерфейс корректно обрабатывает границы
Повторный запуск	Та же допустимая пара данных	Результат не зависит от состояния предыдущего теста

Результаты и обсуждение

В учебном тестовом проекте реализован автоматизированный сценарий положительной аутентификации. Он включает запуск браузера, переход к форме, ввод учётных данных, отправку запроса, ожидание признака успешного перехода и проверку итогового состояния. Сценарий построен без фиксированных задержек Thread.sleep(): выполнение продолжается после наступления условия либо завершается диагностируемым исключением по тайм-ауту.

Таблица 2. Этапы выполнения автоматизированного UI-теста

Этап	Контролируемое условие	Диагностическое значение
Подготовка	Браузер запущен, форма открыта	Исключает влияние неизвестного начального состояния
Ввод данных	Поля видимы и доступны для ввода	Отделяет ошибку локатора от ошибки авторизации
Отправка	Кнопка доступна для нажатия	Предотвращает действие до готовности интерфейса
Ожидание результата	Наступил признак успеха или отказа	Связывает действие с фактическим переходом
Проверка	Фактическое состояние соответствует ожидаемому	Формирует однозначный результат теста
Завершение	Ресурсы браузера освобождены	Обеспечивает независимость повторного запуска

Предложенная матрица расширяет положительный тест негативными и граничными проверками без дублирования общего сценария. Изменяются данные и ожидаемое состояние, тогда как открытие формы, заполнение полей и отправка остаются общими. При росте набора страниц Selenium рекомендует отделять представление страницы от тестов с помощью Page Object Model [3]. В небольшой учебной реализации это направление развития, а в расширенном наборе тестов — способ уменьшить повторение локаторов и действий.

Практическая ценность формализации состоит в воспроизводимости. Инструкция «войти в систему» не задаёт критерия готовности страницы и не объясняет причину сбоя. Кортеж T фиксирует входные данные и ожидаемый результат, а условие W связывает пользовательское действие с наблюдаемым переходом интерфейса. При неуспешном запуске отчёт показывает, какое состояние не было достигнуто до истечения тайм-аута.

Ограничением работы является реализация одного положительного сценария. Для полного покрытия необходимы негативные проверки, восстановление пароля, блокировка учётной записи, разграничение ролей и сетевые ошибки. Дополнительным направлением является параметризация тестов средствами JUnit 5 и запуск в нескольких браузерах.

Заключение

Разработана сценарная методика автоматизированной проверки аутентификации веб-приложения. Тест формализован через предусловия, данные, действия, ожидания и наблюдаемый результат. Синхронизация выполняется явным ожиданием состояния интерфейса, а независимость запусков обеспечивается контролируемой подготовкой и завершением теста. На Java с использованием Selenium WebDriver и JUnit 5 реализован положительный UI-тест и сформирована матрица последующих негативных и граничных сценариев. Методика может служить основой расширяемого набора регрессионных проверок формы входа.

Литература:

1. Selenium. Waiting Strategies. URL: <https://www.selenium.dev/documentation/webdriver/waits/> (дата обращения: 18.07.2026).
2. JUnit Team. JUnit User Guide. URL: <https://junit.org/junit5/docs/current/user-guide/> (дата обращения: 18.07.2026).
3. Selenium. Page object models. URL: https://www.selenium.dev/documentation/test_practices/encouraged/page_object_models/ (дата обращения: 18.07.2026).
4. W3C. WebDriver. URL: <https://www.w3.org/TR/webdriver2/> (дата обращения: 18.07.2026).
5. ISTQB. Certified Tester Foundation Level Syllabus v4.0. URL: <https://www.istqb.org/certifications/certified-tester-foundation-level> (дата обращения: 18.07.2026).
6. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. Hoboken: Wiley, 2011. 256 p.

Влияние качества фильтрации воздуха на отказоустойчивость серверного оборудования и срок его службы

Гаас Даниил Андреевич, студент магистратуры
Национальный исследовательский университет «Московский институт электронной техники» (г. Зеленоград)

Гаас Сергей Андреевич, начальник сектора
ООО «АйЭмТи» (г. Зеленоград)

Статья посвящена влиянию качества фильтрации воздуха на отказоустойчивость серверного оборудования и его элементов, а также влиянию на срок службы оборудования. Рассмотрены методические рекомендации, а также воздействие на оборудование при ненормируемых параметрах воздуха.

Ключевые слова: вентиляция, центр обработки данных, качество воздуха, класс фильтрации, фильтры для вентиляции, параметры воздуха, отказоустойчивость серверного оборудования

Современные центры обработки данных (далее ЦОД) характеризуются высокой плотностью размещения вычислительного и телекоммуникационного оборудования в помещении, и такое размещение приводит к значительному тепловыделению аппаратуры, а соответственно повышается ее тепловая нагрузка и повышается ее чувствительность к отклонениям параметров микроклимата помещения, в том числе запыленности воздуха. В условиях ежедневной эксплуатации и высокой стоимости простоев важной задачей климатического контроля является обеспечение соблюдения нормируемых параметров температуры, влажности и концентрации взвешенных частиц в пределах рекомендованных диапазонов, определяемых отраслевыми стандартами и требованиями производителей оборудования. Так, для серверных залов крупных ЦОД согласно методическим рекомендациям по проектированию центров обработки данных на рисунке 1 представлены оптимальные и допустимые параметры воздуха [1]:

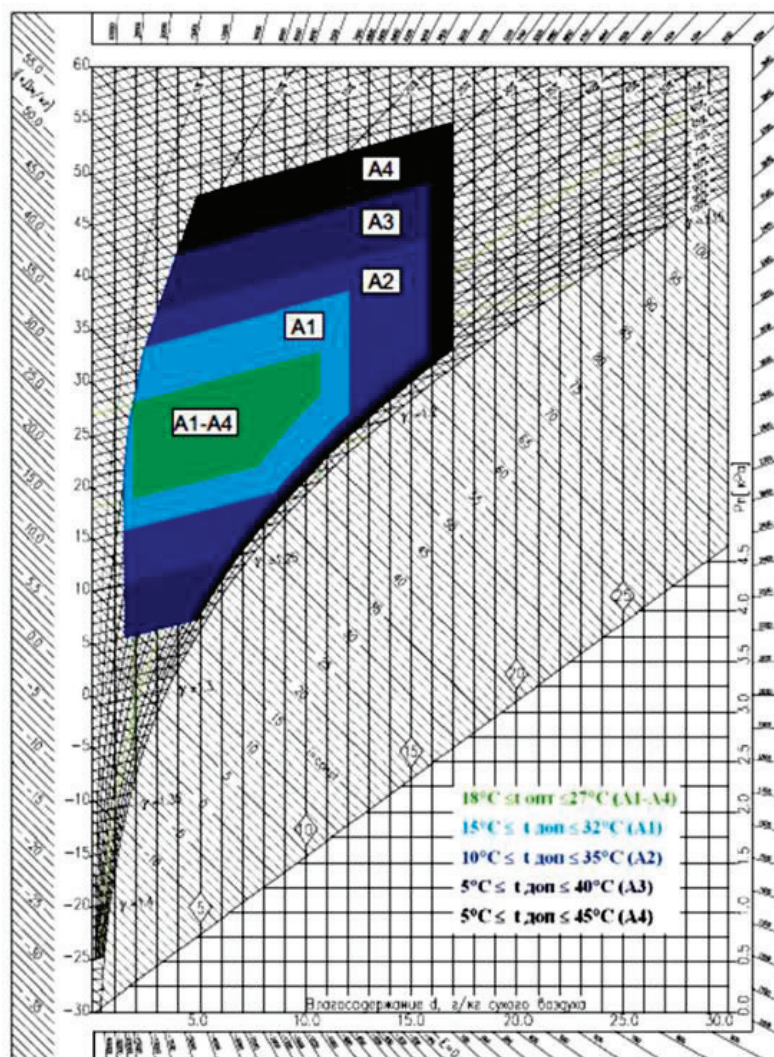


Рис. 1. Диаграмма I-d с диапазонами оптимальных и допустимых параметров воздуха

А также параметры приведены в таблице 1.

Таблица 1. Оптимальные и допустимые параметры воздуха

Параметры воздуха	Класс	Тип ИТ-оборудования	Диапазон температур по сухому термометру, $T_{с.т.}, ^\circ\text{C}$	Диапазон влажности воздуха, $\phi, \%$	Диапазон температур точек росы, $T_p, ^\circ\text{C}$	Максимальная температура точки росы, $T_{p \text{ макс}}, ^\circ\text{C}$
Оптимальные (рекомендуемые)	A1 — A4	Все	18–27	Не выше 60	-9–15	-
Допустимые	A1	Серверы масштаба крупного предприятия, системы хранения данных	15–32	8–80	-12–17	17
	A2	Серверы, системы хранения данных, персональные компьютеры, рабочие станции	10–35	8–80	-12–21	21
	A3		5–40	8–85	-12–24	24
	A4		5–45	8–85	-12–24	24

А также запыленность воздуха населенного пункта должна составлять не более $1,0 \text{ мг/м}^3$. Для такого соблюдения параметров указывается необходимость многоступенчатой очистки воздуха, которая обеспечивает достижение от пятого до восьмого класса чистоты согласно ISO 14644-1. Это напрямую связано с выбором класса фильтра и схем их включения в общеобменную систему вентиляции. Фильтрация воздуха в ЦОД опирается на классификацию фильтрующих элементов, которая предусматривает разделение на грубую, тонкую и высокоэффективную очистку, которая также соответствует классам G, F, HEPA (High Efficiency Particulate Air или High Efficiency Particulate Arrestance — «высокоэффективное удержание частиц в воздухе»). Предусматривается двух- или трехступенчатая схема фильтрации:

- первая ступень отвечает за задержание крупных частиц
- вторая ступень отвечает за тонкую очистку от мелкодисперсной пыли
- в третьей ступени применяются дополнительно HEPA — фильтры, которые соответствуют классу очистки H13 — H14.

Такая архитектура позволяет обеспечивать ограничение концентрации частиц размером 0,3–0,5 мкм и более до уровня, соответствующего классам чистых помещений, что существенно снижает вероятность накопления электростатической активной пыли на поверхности плат, радиаторов и контактных групп. Внешний вид HEPA — фильтра представлен на рисунке 2:



Рис. 2. HEPA-фильтр

Одновременно класс фильтрации наружного воздуха обычно задается не ниже F7 или эквивалентных значений по другим стандартам, поскольку именно на этой стадии удаляются основные массовые фракции пыльцы, волокон и аэрозолей, которые в противном случае приводили бы к ускоренному засорению последующих ступеней и теплообменников.

Запыленность воздуха в серверных помещениях оказывает комплексное воздействие на электронное и электромеханическое оборудование, что в свою очередь влечет за собой ухудшение охлаждения и рост механического износа. Пыль, которая оседает на радиаторах и воздуховодах внутри серверных блоков, снижает эффективность теплоотвода. Это также приводит к увеличению термического сопротивления и способствует росту температуры компонентов, что в свою очередь вызывает ускоренное старение полупроводниковых элементов. Также в условиях повышенной влажности пылевая масса на поверхностях плат уменьшает сопротивление изоляции, вызывает токи утечки и изменяет распределение электрических полей, за которыми следуют локальные перегревы, ложные срабатывания и деградация микросхем. Пыль в механизмах вентилятора соответственно ухудшает качество электрического контакта, усиливает абразивный износ подшипников и движущихся частей, что сокращает ресурс вентиляторных модулей и вспомогательных устройств систем охлаждения [2].

Влияние качества фильтрации воздуха на отказоустойчивость серверного оборудования проявляется через изменение вероятности возникновения отказов, связанных с запыленностью, а также через косвенное воздействие на температурный режим и нагрузку на системы охлаждения. При недостаточной эффективности фильтрации помещения увеличение концентрации пылевых частиц приводит к ускоренному загрязнению внутренних каналов охлаждения серверов и кондиционеров, что повышает аэродинамическое сопротивление, снижает расход воздуха и вызывает рост температур в тепловых зонах, увеличивая риск отказа и снижения производительности троттлинга процессоров [3]. Троттлинг — это защитный механизм, который автоматически снижает производительность процессора или другого компонента, когда тот приближается к критическому уровню нагрева. Его главная задача — это предотвратить перегрев и термическое повреждение чипа. Одновременное повышение запыленности ускоряет процесс токопроводящих мостиков и коррозионных очагов на платах, которые также увеличивают вероятность отказов по электрическим при-

чинам. В дата-центрах, где проводится регламентная замена фильтров, отмечается возможности эксплуатации серверов в течении нескольких лет без необходимости вскрытия корпусов для очистки от пыли, что косвенно свидетельствует о влиянии качество воздуха на снижение объема обслуживающих работ и продлении срока службы оборудования.

Также при высоком уровне очистке воздуха от взвешенных частиц существенно снижается интенсивность отложений пыли на теплообменных поверхностях, контактах изоляции, что уменьшает скорость износа компонентов и позволяет эксплуатировать оборудование в пределах расчетного ресурса без существенного повышения частоты отказов. В то же время недостаточный класс фильтрации либо временная замена фильтров приводит к росту запыленности, ускоренному старению материалов и контактов, увеличению нагрузки на вентиляторы и кондиционеры. С этим связана рекомендация рассматривать систему фильтрации воздуха как один из ключевых элементов обеспечения жизненного цикла ЦОД, включающий не только подбор классов фильтров, но и организацию мониторинга их состояния и регламентов обслуживания. Классы фильтров оказывают прямое влияние на энергетическую эффективность системы вентиляции, которое также необходимо учитывать при анализе отказоустойчивости и срока службы аппаратуры. Фильтры более высокого класса очистки обладают увеличенным аэродинамическим сопротивлением, которое требует большей мощности вентиляторов и повышает энергопотребление, особенно при загрязнении фильтрующего материала, однако их применение позволяет снизить запыленность до уровней, при которых риск отказов и дополнительная нагрузка на оборудование минимизируются [4]. В этом смысле оптимизация схемы фильтрации сводится к выбору комбинаций ступеней очистки и классов фильтров, обеспечивающей требуемый класс чистоты при приемлемом уровне энергозатрат и сложности обслуживания, что может быть реализовано посредством анализа сопротивления, запыленности и срока службы фильтров с учетом реальных условий эксплуатации ЦОД.

Таким образом, качество фильтрации воздуха, выражаемое через выбор классов фильтров, конфигурацию схем очистки, режимы эксплуатации и обслуживания, оказывает комплексное и многофакторное влияние на отказоустойчивость серверного оборудования и срок его службы в условиях центров обработки данных. Обеспечение соответствия воздушной среды машинных залов требованиям стандартов чистых помещений и рекомендациям производителей, организация многоступенчатой фильтрации и систем мониторинга состояния фильтров позволяют существенно снизить запыленность, минимизировать тепловые и электрические риски и обеспечить эксплуатацию серверной инфраструктуры в пределах расчётного ресурса без существенного роста частоты отказов.

Литература:

1. Методические рекомендации по проектированию центров обработки данных / Министерство строительства и жилищно-коммунального хозяйства Российской Федерации, Федеральный центр нормирования, стандартизации и технической оценки соответствия в строительстве // URL: <https://meganorm.ru/Data2/1/4293721/4293721780.pdf> (дата обращения: 07.07.26).
2. Коротков А. Д. Влияние вентиляции на надёжность оборудования машинного зала энергоцентра // Актуальные исследования. 2026. № 27 (313). Ч. I. С. 48–51. URL: <https://apni.ru/article/15641-vliyanie-ventilyacii-na-nadyozhnost-oborudovaniya-mashinnogo-zala-energocentra>. (дата обращения: 10.07.26).
3. Вентиляция в серверной: нормы кратности воздухообмена и проектирование систем // АйТитело URL: <https://www.ittelo.ru/news/ventilyatsiya-v-servernoy-normy-kratnosti-vozdukhoobmena-i-proektirovanie-sistem/> (дата обращения: 10.07.26).
4. Вентиляция серверных и IT-комнат: проектирование, монтаж и эксплуатация // ООО «Вентстрой» URL: <https://airovent.ru/articles/ventilyatsiya-servernykh-i-it-komnat-proektirovanie-montazh-i-ekspluatatsiya/> (дата обращения: 15.07.26).

Эволюция систем предотвращения вторжений: от сигнатурного анализа к поведенческим методам и машинному обучению

Галкин Александр Алексеевич, студент
Пензенский государственный технологический университет

В статье рассматривается эволюция систем предотвращения вторжений (IPS) — от первых сигнатурных решений до современных интеллектуальных платформ, использующих методы машинного обучения. Проанализированы ключевые этапы развития: сигнатурный анализ, поведенческий анализ и выявление аномалий, глубокий анализ пакетов, интеграция с источниками Threat Intelligence. Отдельное внимание уделено современным технологическим трендам —

применению искусственного интеллекта, адаптивным политикам безопасности, контекстному анализу и защите облачных инфраструктур, а также перспективным направлениям научных исследований в данной области.

Ключевые слова: информационная безопасность, системы предотвращения вторжений, IPS, машинное обучение, поведенческий анализ, обнаружение аномалий, искусственный интеллект.

Развитие цифровых технологий, широкое распространение облачных вычислений, Интернета вещей (IoT), промышленного Интернета вещей (IIoT), мобильных платформ и распределённых вычислительных инфраструктур привело к резкому увеличению количества и сложности киберугроз [1]. Современные информационные системы функционируют в условиях постоянно изменяющегося ландшафта угроз, где злоумышленники активно используют автоматизацию, технологии искусственного интеллекта, вредоносные программы нового поколения и методы сокрытия своей деятельности. В этих условиях традиционные средства защиты уже не способны обеспечить необходимый уровень безопасности.

Особое место среди средств сетевой защиты занимают системы предотвращения вторжений (Intrusion Prevention Systems, IPS). Их основная задача заключается не только в обнаружении признаков компьютерной атаки, но и в активном противодействии ей посредством блокирования вредоносного трафика, разрыва соединений, изменения правил фильтрации и выполнения других защитных действий в режиме реального времени.

За более чем два десятилетия своего развития IPS прошли значительную эволюцию. Первоначально они представляли собой сравнительно простые системы, основанные на поиске заранее известных сигнатур атак. Однако постоянное появление новых методов компрометации информационных систем потребовало создания более интеллектуальных механизмов анализа сетевого трафика. Постепенно в IPS были внедрены методы анализа поведения пользователей и приложений, статистические модели выявления аномалий, технологии корреляции событий безопасности и алгоритмы машинного обучения.

В статье рассмотрены основные этапы эволюции IPS, современные технологические тенденции, а также перспективные направления научных исследований в области интеллектуальных систем предотвращения вторжений.

Этапы развития IPS

Сигнатурный анализ

Первые поколения IPS были основаны на сигнатурном подходе. Принцип их работы заключался в сравнении сетевых пакетов или последовательностей команд с заранее сформированными шаблонами известных атак. Каждая сигнатура представляла собой описание конкретной угрозы: характерной последовательности байтов, команды, заголовка сетевого пакета или специфического поведения сетевого протокола.

Преимуществом сигнатурного анализа являлась высокая точность обнаружения хорошо изученных атак. Однако данный подход имел существенные ограничения: IPS могла обнаружить только те угрозы, информация о которых уже присутствовала в базе знаний. Особенно остро проблема проявилась с появлением атак нулевого дня (Zero-Day), использующих ранее неизвестные уязвимости, а также технологий полиморфизма и обфускации вредоносного кода [1].

Поведенческий анализ и выявление аномалий

Следующим этапом развития стало внедрение методов поведенческого анализа. В отличие от сигнатурного подхода такие системы ориентировались не на поиск известных шаблонов, а на построение модели нормального функционирования информационной системы. Для этого анализировались статистические характеристики сетевого трафика, интенсивность обмена данными, последовательность действий пользователей и особенности функционирования приложений [2].

Такой подход позволил обнаруживать ранее неизвестные угрозы, включая сложные целевые атаки (Advanced Persistent Threats, APT) и внутренние угрозы. Однако эффективность поведенческого анализа напрямую зависела от качества построенной модели нормального поведения: изменения бизнес-процессов, обновление программного обеспечения или переход сотрудников на удалённую работу могли восприниматься системой как аномалии, увеличивая количество ложных срабатываний.

Контроль приложений и анализ сетевых протоколов

Развитие сетевых сервисов привело к тому, что многие современные приложения начали использовать нестандартные порты и зашифрованные соединения, что существенно осложнило применение традиционных методов фильтрации. В ответ IPS получили возможность анализа приложений независимо от используемых транспортных про-

токолов, а также технологии глубокого анализа пакетов (Deep Packet Inspection, DPI), позволяющие исследовать содержимое сетевого обмена вплоть до уровня прикладных протоколов.

Интеграция с внешними источниками данных

Следующим этапом эволюции стало использование внешних источников информации об угрозах. Современные IPS активно взаимодействуют с системами Threat Intelligence, получая сведения о новых вредоносных IP-адресах, доменах, хэшах файлов и индикаторах компрометации (IOC). Подобная интеграция позволяет значительно сократить время реакции на новые угрозы и повысить достоверность принимаемых решений.

Современные технологические тренды

Машинное обучение и искусственный интеллект

Одним из наиболее перспективных направлений развития IPS является применение методов машинного обучения и искусственного интеллекта. Современные алгоритмы способны автоматически анализировать огромные массивы сетевого трафика, выявляя сложные закономерности, которые трудно обнаружить традиционными методами [1, 3]. Используются как методы обучения с учителем, основанные на заранее размеченных наборах данных, так и методы обучения без учителя, позволяющие самостоятельно обнаруживать аномалии; особое распространение получили алгоритмы кластеризации, деревья решений, случайные леса, методы опорных векторов и нейронные сети [3].

Вместе с тем использование ИИ порождает новые вызовы: для обучения моделей требуются большие объёмы качественных данных, а сами модели могут быть чувствительны к изменению характеристик защищаемой среды или подвергаться атакам на алгоритмы машинного обучения.

Адаптивные политики безопасности

Современные IPS постепенно переходят от статических правил к динамическому управлению политиками безопасности. Такие системы способны самостоятельно изменять уровень контроля в зависимости от текущей ситуации: уровня угрозы, времени суток, состояния информационной инфраструктуры или критичности защищаемых ресурсов. Это особенно значимо для защиты критической инфраструктуры, где цена ложного срабатывания или пропуска атаки крайне высока [4].

Контекстный анализ

Современные решения всё чаще принимают решение не только на основании анализа одного сетевого пакета, но и с учётом дополнительной информации: репутации IP-адреса, истории поведения пользователя, уровня доверия к устройству, результатов аутентификации и данных о предыдущих инцидентах. Развитием этого направления являются provenance-based системы, восстанавливающие полную цепочку событий для точной атрибуции атаки [5]. Комплексный анализ множества факторов значительно повышает качество обнаружения атак и снижает вероятность ошибочной блокировки легитимной деятельности.

Защита облачных и гибридных инфраструктур

Современные организации всё чаще используют гибридные модели размещения информационных систем, объединяя локальные центры обработки данных с публичными облаками и контейнерными платформами. В таких условиях классические сетевые IPS уже не способны обеспечить полноценный контроль всего информационного обмена, что стимулирует развитие виртуальных IPS, средств защиты контейнеров и облачных платформ предотвращения вторжений, интегрированных с архитектурами Zero Trust и SASE.

Перспективные направления научных исследований

Одним из наиболее актуальных направлений является развитие объяснимого искусственного интеллекта (Explainable Artificial Intelligence, XAI): для специалистов по информационной безопасности крайне важно понимать причины, по которым интеллектуальная система приняла решение о блокировке того или иного сетевого события.

Другим перспективным направлением является создание самообучающихся IPS, способных непрерывно адаптировать свои модели без участия администратора, а также прогнозирование атак на основе анализа временных рядов

и графов взаимодействия объектов. Не менее важным направлением становится интеграция IPS с платформами киберразведки и использование больших языковых моделей (LLM) для автоматического анализа журналов безопасности и поддержки специалистов центров мониторинга (SOC). Ещё одним направлением является создание федеративных моделей машинного обучения, позволяющих нескольким организациям совместно обучать алгоритмы обнаружения угроз без передачи конфиденциальных данных.

Заключение

Эволюция систем предотвращения вторжений отражает фундаментальные изменения, происходящие в современной информационной безопасности. Если первые IPS представляли собой инструменты обнаружения заранее известных угроз, то современные решения становятся интеллектуальными платформами, объединяющими сигнатурный анализ, поведенческое моделирование, методы искусственного интеллекта, анализ контекста и данные киберразведки. Для научного сообщества данная область остаётся одной из наиболее перспективных: исследования в сфере интеллектуального обнаружения вторжений, объяснимого ИИ, адаптивных моделей безопасности и федеративного обучения будут определять развитие технологий защиты информации в ближайшие годы.

Литература:

1. Momand A., Jan S. U., Ramzan N. A Systematic and Comprehensive Survey of Recent Advances in Intrusion Detection Systems Using Machine Learning: Deep Learning, Datasets, and Attack Taxonomy // Journal of Sensors. — 2023. — Vol. 2023. — Art. 6048087. — URL: <https://doi.org/10.1155/2023/6048087>.
2. Awajan A. A Novel Deep Learning-Based Intrusion Detection System for IoT Networks // Computers. — 2023. — Vol. 12, No. 2. — P. 34. — URL: <https://doi.org/10.3390/computers12020034>.
3. Ali A. H., Charfeddine M., Ammar B., Hamed B. B., Albalwy F., Alqarafi A., Hussain A. Unveiling Machine Learning Strategies and Considerations in Intrusion Detection Systems: A Comprehensive Survey // Frontiers in Computer Science. — 2024. — Vol. 6. — Art. 1387354. — URL: <https://doi.org/10.3389/fcomp.2024.1387354>.
4. Pinto A., Herrera L.-C., Donoso Y., Gutierrez J. A. Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure // Sensors. — 2023. — Vol. 23, No. 5. — Art. 2415. — URL: <https://doi.org/10.3390/s23052415>.
5. Zipperle M., Gottwalt F., Chang E., Dillon T. S. Provenance-Based Intrusion Detection Systems: A Survey // ACM Computing Surveys. — 2022. — Vol. 55. — P. 1–36. — URL: <https://doi.org/10.1145/3494524>.

Сравнительный анализ методов защиты веб-приложений от SQL-инъекций и XSS-атак

Дунаев Лев Дмитриевич, студент;

Хусяинова Алина Маратовна, студент;

Щавровская Полина Константиновна, студент

Уральский федеральный университет имени первого Президента России Б. Н. Ельцина (г. Екатеринбург)

В статье рассматривается актуальная проблема обеспечения безопасности веб-приложений в контексте двух наиболее распространённых классов уязвимостей — SQL-инъекций (SQL Injection) и межсайтового скриптинга (Cross-Site Scripting, XSS). Цель исследования — сравнительный анализ эффективности и накладных расходов различных методов защиты на учебном экспериментальном стенде. Для SQL-инъекций протестированы три подхода: конкатенация SQL-запросов без защиты, параметризованные запросы (Prepared Statements) и ORM-подход с дополнительной валидацией входных данных. Для XSS исследованы четыре стратегии: отсутствие фильтрации, HTML-экранирование, комбинация sanitization и escape, а также совместное применение Content Security Policy (CSP) и экранирования. Эксперимент проводился на локальном стенде с использованием Python 3.11 и SQLite; для SQL-инъекций применялся набор из 6 типовых payload-ов OWASP, для XSS — 6 репрезентативных векторов атак. Дополнительно измерялась средняя и 95-перцентильная задержка выполнения запросов к базе данных. Результаты показали, что параметризованные запросы и ORM обеспечивают 100 % блокировку SQL-инъекций при минимальном влиянии на производительность. Для XSS наиболее надёжным оказалась комбинация sanitization и HTML-escape; CSP в сочетании с экранированием формирует дополнительный уровень Defense in Depth. На основании эксперимента сформулированы практические рекомендации для разработчиков программных систем.

Ключевые слова: веб-безопасность, SQL-инъекция, XSS, prepared statements, ORM, Content Security Policy, OWASP, программная инженерия.

1. Введение

Веб-приложения занимают центральное место в цифровой инфраструктуре образовательных организаций, государственных сервисов и коммерческих систем. По данным OWASP Top 10 (2021), уязвимости, связанные с некорректной обработкой пользовательского ввода, остаются одной из главных причин компрометации информационных систем [1]. Среди них выделяются SQL-инъекции (A03:2021 — Injection) и XSS (A03:2021, в составе Injection / отдельно в более ранних версиях), которые эксплуатируются как начинающими, так и опытными злоумышленниками.

SQL-инъекция позволяет атакующему модифицировать логику SQL-запроса путём вставки вредоносных фрагментов в поля ввода. Последствия включают несанкционированное чтение, изменение и удаление данных, обход аутентификации и потенциальное компрометирование сервера. XSS, в свою очередь, заключается во внедрении исполняемого JavaScript-кода в контекст web-страницы, что может привести к краже сессионных cookie, фишингу, дефacement и распространению вредоносного кода среди пользователей.

Несмотря на зрелость рекомендаций индустрии (OWASP, NIST, CWE), на практике разработчики нередко применяют устаревшие или неполные механизмы защиты. Особенно это характерно для учебных и MVP-проектов, где приоритет отдаётся скорости разработки, а не безопасности. Для студентов направления «Программная инженерия» критически важно не только знать теоретические определения уязвимостей, но и уметь количественно сравнивать методы защиты.

Цель данной работы — провести сравнительный анализ методов защиты веб-приложений от SQL-инъекций и XSS на воспроизводимом учебном стенде, оценить эффективность блокировки атак и измерить влияние защитных механизмов на производительность. Задачи исследования: (1) обзор теоретических основ и существующих подходов; (2) проектирование экспериментального стенда; (3) формирование тестовых сценариев на основе OWASP; (4) проведение эксперимента и статистическая обработка результатов; (5) формулирование практических рекомендаций для разработчиков.

2. Обзор методов защиты

2.1. Защита от SQL-инъекций

Классификация CWE-89 (Improper Neutralization of Special Elements used in an SQL Command) описывает SQL-инъекцию как следствие включения недоверенных данных в SQL-запрос без должной нейтрализации [2]. Основные стратегии защиты:

1) Параметризованные запросы (Prepared Statements) — разделение структуры запроса и данных; СУБД интерпретирует параметры исключительно как значения, а не как исполняемый код [3].

2) ORM (Object-Relational Mapping) — абстракция доступа к данным через объектную модель; при корректном использовании API ORM также генерирует параметризованные запросы [4].

3) Валидация и whitelist-фильтрация входных данных — дополнительный слой, не заменяющий параметризацию, но снижающий поверхность атаки.

4) Принцип наименьших привилегий для учётной записи БД — ограничение последствий успешной эксплуатации.

Категорически не рекомендуется полагаться на «чёрные списки» (blacklist) запрещённых символов и ручное экранирование кавычек — эти методы обходятся кодированием, вложенными запросами и особенностями конкретной СУБД [1].

2.2. Защита от XSS

XSS (CWE-79) подразделяется на Reflected, Stored и DOM-based [5]. В данной работе исследуется reflected XSS — наиболее наглядный сценарий для учебного стенда. Механизмы защиты:

1) Output encoding (HTML-escape) — преобразование опасных символов (<, >, &, «, ') в HTML-сущности при выводе данных в HTML-контекст [6].

2) Input sanitization — удаление или нейтрализация HTML-тегов и опасных URI-схем (javascript:) на входе.

3) Content Security Policy (CSP) — декларативный механизм браузера, ограничивающий источники исполняемых скриптов и снижающий вероятность успешной эксплуатации даже при наличии инъекции [7].

4) HttpOnly и Secure флаги для cookie — снижение последствий XSS, но не устранение самой уязвимости.

Согласно концепции Defense in Depth, оптимальная стратегия предполагает комбинацию нескольких независимых слоёв защиты [8].

3. Методология исследования

Исследование выполнено методом controlled experiment (контролируемого эксперимента) на локальном учебном стенде. Гипотезы:

H1: параметризованные SQL-запросы и ORM-подход блокируют 100 % тестовых SQL-инъекций при сохранении сопоставимой производительности с незащищённым кодом.

H2: HTML-экранирование блокирует reflected XSS для большинства tag-based payload-ов, но менее эффективно в нестандартных контекстах.

H3: комбинация sanitization, escape и CSP обеспечивает наиболее высокий уровень защиты от XSS.

Независимые переменные: выбранный метод защиты (SQL / XSS). Зависимые переменные: доля заблокированных атак (Block Rate, %), среднее время выполнения запроса (avg, мс), 95-й перцентиль времени (p95, мс). Контролируемые факторы: версия Python 3.11, СУБД SQLite 3, аппаратная платформа исследователя, фиксированный набор payload-ов.

4. Архитектура экспериментального стенда

Стенд реализован на языке Python 3.11 и состоит из двух логических модулей: (1) модуль доступа к данным с тремя режимами выполнения SQL-запросов (конкатенация строк, параметризованные запросы, ORM с валидацией); (2) модуль рендеринга HTML-комментариев с четырьмя режимами обработки вывода (без фильтрации, HTML-экранирование, sanitization + escape, CSP + escape). База данных SQLite содержит таблицу users (id, username, password) с тремя тестовыми записями. Воспроизводимость эксперимента обеспечивается фиксированным набором входных данных и однотипной процедурой измерений; при необходимости стенд может быть повторён в любой среде с Python 3.11 и SQLite без внешних зависимостей.

Для SQL-инъекций использовался набор из 6 payload-ов, рекомендованных OWASP Testing Guide [9]: ' OR '1'='1; admin'—; ' UNION SELECT null, null—; 1; DROP TABLE users—; ' OR 1=1—; ') OR ('1'='1. Критерий успешной атаки: получение более одной записи, обход фильтрации по username или выполнение UNION-запроса.

Для XSS применялся набор из 6 векторов: <script>alert(1)</script>; ; <svg/onload=alert(1)>; javascript:alert(1); <iframe src=javascript:alert(1)>; "><script>alert('XSS')</script>. Критерий успешной атаки: наличие в результирующем HTML неэкранированных опасных тегов или обработчиков событий (onerror, onload и т. д.).

Измерение производительности SQL-режимов выполнялось серией из 200 последовательных запросов с легитимным значением username = 'student'; фиксировались среднее арифметическое и 95-й перцентиль времени отклика.

5. Результаты эксперимента

5.1. Эффективность защиты от SQL-инъекций

Таблица 1. Результаты тестирования методов защиты от SQL-инъекций

Метод защиты	Всего атак	Заблок. атак	Успешн. атак	Block Rate	avg, мс	p95, мс
Raw SQL (без защиты)	6	2	4	33.3 %	0.162	0.222
Prepared Statements	6	6	0	100.0 %	0.159	0.195
ORM + валидация	6	6	0	100.0 %	0.164	0.208

Результаты (табл. 1) подтверждают гипотезу H1. Незащищённый режим Raw SQL пропустил 4 из 6 атак (66,7 % успешных эксплуатаций). Наиболее критичными оказались payload-ы семейства tautology (' OR '1'='1, ' OR 1=1—), позволяющие извлечь все строки таблицы users. Prepared Statements и ORM с валидацией заблокировали все 6 атак (Block Rate = 100 %).

Анализ производительности показал, что параметризация не создаёт существенных накладных расходов: avg = 0,159 мс против 0,162 мс у Raw SQL (разница менее 10 %). Это согласуется с выводами Howard и LeBlanc (2003) о том, что параметризованные запросы должны рассматриваться как стандартная практика, а не как «оптимизация безопасности» [3].

5.2. Эффективность защиты от XSS

Таблица 2. Результаты тестирования методов защиты от XSS

Метод защиты	Всего атак	Заблок. атак	Успешн. атак	Block Rate
Без фильтрации	6	1	5	16.7 %
HTML-экранирование	6	6	0	100.0 %
Sanitization + escape	6	6	0	100.0 %
CSP + escape	6	6	0	100.0 %

Без фильтрации успешно эксплуатировались 5 из 6 векторов (83,3 % успешных атак). Payload javascript:alert(1) в текстовом контексте <div> не формирует исполняемый HTML-элемент и по выбранному критерию не считается успешной DOM-эксплуатацией, однако остаётся потенциально опасным в контексте атрибутов href и src.

HTML-экранирование, sanitization+escape и CSP+escape продемонстрировали Block Rate 100 % для тестового набора reflected XSS. Между HTML-escape и sanitization+escape выбор зависит от контекста: escape достаточен при строгом выводе в HTML body, тогда как sanitization необходим при допустимости ограниченного подмножества HTML-разметки (rich text). CSP выступает резервным барьером: даже при ошибке серверной фильтрации браузер блокирует inline-скрипты при корректной политике script-src 'none' или nonce-based политике [7].

5.3. Сводная оценка методов

Таблица 3. Сводная оценка методов защиты

Угроза	Рекомендуемый метод	Эффективность	Overhead	Приоритет внедрения
SQL Injection	Prepared Statements	100 %	Низкий	Обязательный
SQL Injection	ORM (SQLAlchemy, Django ORM)	100 %*	Низкий	Обязательный
SQL Injection	Raw SQL + конкатенация	33 %	Минимальный	Запрещён
Reflected XSS	HTML-escape при выводе	100 %**	Минимальный	Обязательный
Reflected XSS	Sanitization + escape	100 %**	Низкий	Для rich text
Reflected XSS	CSP + escape	100 %**	Низкий	Defense in Depth
Любая XSS	Только blacklist фильтрация	Нестабильная	Низкий	Недостаточно

* при условии использования параметризованного API ORM;

** для тестируемого набора reflected XSS в HTML body-контексте.

6. Обсуждение результатов

Полученные данные согласуются с индустриальными рекомендациями OWASP ASVS (Application Security Verification Standard) уровня 2, предписывающими использование параметризованных интерфейсов доступа к данным и контекстного output encoding [10]. Эксперимент наглядно демонстрирует, что отказ от конкатенации SQL-строк не требует компромисса в производительности для типичных CRUD-операций учебных и малых production-систем.

Ограничения исследования: (1) использована одна СУБД (SQLite); поведение PostgreSQL/MySQL может отличаться для edge-case payload-ов; (2) не исследованы stored и DOM-based XSS; (3) не оценивался WAF (Web Application Firewall) как внешний периметровый слой; (4) стенд не моделирует конкурентную нагрузку. Дальнейшие исследования могут включать расширение payload-набора до OWASP SQLi/XSS cheat sheet, тестирование на PostgreSQL и интеграцию статического анализатора (Semgrep, Bandit) для выявления уязвимостей на этапе CI/CD.

7. Практические рекомендации для разработчиков

На основании проведённого эксперимента предлагаются следующие рекомендации для команд программной инженерии:

1. Никогда не формировать SQL-запросы конкатенацией строк с пользовательским вводом; использовать параметризованные запросы или ORM.
2. Применять HTML-escape на этапе вывода (а не только фильтрацию на входе) для всех динамических данных в HTML-контексте.

3. Внедрять CSP как дополнительный слой; минимальная политика: default-src 'self'; script-src 'self' или nonce-based.
4. Включать SAST/DAST проверки в pipeline непрерывной интеграции.
5. Проводить регулярный security review кода, включая учебные проекты — формирование secure coding habits на 3-м курсе снижает риск уязвимостей в промышленной разработке.

8. Заключение

В работе выполнен сравнительный анализ методов защиты веб-приложений от SQL-инъекций и XSS на воспроизводимом экспериментальном стенде. Установлено, что конкатенация SQL-запросов блокирует лишь часть атак и не может применяться в production-системах. Prepared Statements и ORM обеспечивают полную защиту от тестового набора SQL-инъекций без существенного снижения производительности. Для reflected XSS эффективны HTML-экранирование, sanitization и CSP; наиболее устойчивая стратегия — многоуровневая (escape + CSP). Результаты могут быть использованы в учебных курсах по веб-разработке, безопасности программного обеспечения и при подготовке проектной документации студентов направления 09.03.04 «Программная инженерия».

Литература:

1. OWASP Top 10–2021: The Ten Most Critical Web Application Security Risks. — URL: <https://owasp.org/Top10/> (дата обращения: 12.06.2026).
2. CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection'). — URL: <https://cwe.mitre.org/data/definitions/89.html> (дата обращения: 12.06.2026).
3. Howard M., LeBlanc D. Writing Secure Code. — 2nd ed. — Microsoft Press, 2003. — 768 p.
4. Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2002. — 560 p.
5. CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'). — URL: <https://cwe.mitre.org/data/definitions/79.html> (дата обращения: 15.06.2026).
6. OWASP XSS Prevention Cheat Sheet. — URL: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (дата обращения: 15.06.2026).
7. OWASP Content Security Policy Cheat Sheet. — URL: https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html (дата обращения: 17.06.2026).
8. NIST SP 800–53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations. — NIST, 2020.
9. OWASP Web Security Testing Guide v4.2. — URL: <https://owasp.org/www-project-web-security-testing-guide/> (дата обращения: 18.06.2026).
10. OWASP Application Security Verification Standard (ASVS) 4.0.3. — URL: <https://owasp.org/www-project-application-security-verification-standard/> (дата обращения: 19.06.2026).
11. Stuttard D., Pinto M. The Web Application Hacker's Handbook. — 2nd ed. — Wiley, 2011. — 912 p.
12. McGraw G. Software Security: Building Security In. — Addison-Wesley, 2006. — 448 p.
13. ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems.
14. Saltzer J. H., Schroeder M. D. The Protection of Information in Computer Systems // Proceedings of the IEEE. — 1975. — Vol. 63, № 9. — P. 1278–1308.
15. Bannikov A. Yu. et al. Secure Software Development Lifecycle Guidelines // Proc. of IEEE Security Workshops. — 2019.

Микрофронтенд-архитектура: понятие и предпосылки возникновения

Жарков Никита Александрович, студент
Московский государственный технологический университет «СТАНКИН»

В статье вводится определение микрофронтенд-архитектуры как подхода к разработке клиентской части приложений, основанного на принципах микросервисов. Рассматриваются исторические предпосылки возникновения — кризис монолитных фронтенд-приложений, необходимость независимой работы команд и ускорения циклов релизов. Описаны ключевые принципы: технологическая агностичность, автономность команд, явные контракты и изоляция ошибок. Приводятся базовые преимущества и даётся краткий обзор ситуаций, в которых микрофронтенды оправданы.

Введение

Современные веб-приложения перестали быть простыми наборами статических страниц. Они превратились в сложные программные системы с сотнями экранов, десятками интеграций и многомиллионной аудиторией. Разработка таких систем сталкивается с проблемами, которые ранее были характерны только для бэкенда: разрастание кодовой базы, длительные циклы сборки, сложность координации множества разработчиков. В ответ на эти вызовы возникла микрофронтенд-архитектура — подход, переносящий идеи микросервисов на клиентскую сторону.

Термин «micro frontends» был популяризирован в 2016 году компанией ThoughtWorks в Technology Radar. С тех пор он прошёл путь от экспериментальной техники до рекомендации к активному применению. Сегодня микрофронтенды используются в таких компаниях, как Microsoft, Spotify, Upwork, Allegro, Leroy Merlin. Цель данной статьи — дать базовое понимание сути подхода, причин его возникновения и основных принципов.

Понятие микрофронтенд-архитектуры

Микрофронтенд — это архитектурный стиль, при котором клиентское приложение разбивается на независимо разрабатываемые, тестируемые и развёртываемые части, каждая из которых отвечает за отдельную предметную область или функциональный блок интерфейса. По определению Мартина Фаулера, «микрофронтенды — это подход к разработке фронтенда, при котором отдельные части интерфейса разрабатываются разными командами с использованием разных технологий и поставляются как самостоятельные приложения».

В практическом смысле микрофронтенд — это отдельное веб-приложение, которое может иметь собственную систему маршрутизации, управление состоянием, дизайн-систему и пайплайн развёртывания. Несколько таких приложений объединяются в единый пользовательский интерфейс через механизмы интеграции на стороне клиента или сервера.

Предпосылки возникновения

До появления микрофронтендов большинство крупных веб-приложений строилось как монолиты: единая кодовая база, единая технологическая платформа, единый процесс релиза. Такой подход имеет серьёзные недостатки, которые становятся критическими по мере роста проекта:

- Единая точка сборки и развёртывания. Любое изменение, даже в одном компоненте, требует пересборки и переразвёртывания всего приложения. Это замедляет циклы релизов и увеличивает риски.

- Жёсткая технологическая связка. Выбор фреймворка (Angular, React, Vue) диктуется на старте и сохраняется на годы, что затрудняет обновления и миграцию на новые версии.

- Невозможность разделить ответственность. Когда десятки разработчиков правят один репозиторий, неизбежны конфликты слияния, регрессии и сложность код-ревью.

- Сложность масштабирования команд. Процесс онбординга новых разработчиков становится трудоёмким из-за большого объёма кода и жёстких связей между модулями.

Эти проблемы хорошо известны бэкэнд-разработчикам, которые решили их с помощью микросервисов. Закономерным шагом стало применение аналогичного подхода на клиентской стороне. Дополнительным драйвером стало широкое распространение практик DevOps и непрерывной поставки (CI/CD), которые требуют возможности независимого релиза компонентов.

Ключевые принципы

В основе микрофронтенд-архитектуры лежат несколько фундаментальных принципов:

Технологическая агностичность — каждая команда может выбирать свой стек технологий (фреймворк, библиотеки, инструменты сборки) и обновлять его независимо от остальных. Это не означает, что все части должны быть написаны на разных фреймворках, но такая возможность существует и особенно полезна при постепенной миграции с устаревших решений.

Независимость команд — команды организуются вокруг вертикальных бизнес-слоёв (например, страница корзины, профиль пользователя, каталог товаров), а не вокруг технических слоёв (UI, бизнес-логика, данные). Каждая команда владеет всем стеком от данных до интерфейса в своей области.

Автономность развёртывания — каждый микрофронтенд может быть собран и развёрнут в любое время без координации с другими частями. Это ускоряет доставку новых функций и упрощает откат изменений.

Явные контракты — взаимодействие между микрофронтендами строится на чётко определённых интерфейсах: публичные API, события, маршруты, дизайн-токены. Это позволяет сохранять слабую связанность и избегать неявных зависимостей.

Изоляция ошибок — сбой в одном микрофронтенде (например, падение JavaScript) не должен приводить к краху всего приложения. Это требует грамотной стратегии загрузки, обработки ошибок и fallback-поведения.

Преимущества подхода

Следование этим принципам даёт ряд существенных выгод:

- ускорение разработки за счёт параллельной работы команд;

- сокращение времени релиза (time-to-market) благодаря независимому деплою;

- возможность постепенного обновления технологического стека;

- упрощение тестирования и отладки за счёт меньшего размера отдельных модулей;
- повышение отказоустойчивости за счёт изоляции.

Заключение

Микрофронтенд-архитектура — это закономерный ответ на вызовы современной фронтенд-разработки. Она

позволяет разделить сложную систему на управляемые части, дать командам свободу выбора технологий и ускорить поставку ценности конечным пользователям. Однако этот подход требует дисциплины в определении границ, согласовании контрактов и организации команд. Понимание базовых принципов, изложенных в данной статье, является необходимым фундаментом для дальнейшего изучения более глубоких аспектов микрофронтендов.

Литература:

1. Cam, Jackson Micro Frontends / Jackson Cam. — Текст: электронный // martinFowler: [сайт]. — URL: <https://martinfowler.com/articles/micro-frontends.html> (дата обращения: 14.07.2026).
2. Evan, Carter Micro-Frontends: Are They Still Worth It in 2025? / Carter Evan. — Текст: электронный // FeatureSlicedDesign: [сайт]. — URL: <https://feature-sliced.design/ru/blog/micro-frontend-architecture> (дата обращения: 14.07.2026).
3. Stephen, Watts An Introduction to Micro Frontends / Watts Stephen. — Текст: электронный // BMC Blogs: [сайт]. — URL: <https://blogs.bmc.com/micro-frontends/> (дата обращения: 14.07.2026).
4. Madhu, Rebbana Micro Frontend Architecture In Enterprise Applications: Benefits And Challenges / Rebbana Madhu. — Текст: электронный // JournalOf: [сайт]. — URL: <https://jicrcr.com/index.php/jicrcr/article/view/3398> (дата обращения: 14.07.2026).

Сравнительный анализ и ограничения микрофронтенд-архитектуры

Жарков Никита Александрович, студент

Московский государственный технологический университет «СТАНКИН»

В статье проводится сравнение микрофронтенд-архитектуры с традиционными подходами. Выделяются основные ограничения и сложности внедрения: проблема производительности, UI/UX-расходимость, трудности интеграционного тестирования. Обсуждаются организационные предпосылки успеха — переход к автономным продуктовым командам и внедрения DevOps-культуры.

Введение

Микрофронтенды привлекают внимание многих организаций, однако их внедрение сопряжено с рядом компромиссов. То, что даёт преимущества в одной плоскости (автономия команд, независимый деплой), порождает сложности в другой (производительность, согласованность интерфейса, управление зависимостями). В данной статье мы сравним микрофронтенды с альтернативными подходами, подробно разберём их ограничения и систематизируем типичные ошибки, возникающие при использовании этой архитектуры.

Сравнение с монолитом

Монолитный фронтенд — это единое приложение, которое собирается из одной кодовой базы и развёртывается целиком. Его преимущества: простота разработки на ранних этапах, отсутствие накладных расходов на интеграцию, лёгкость сквозного тестирования. Однако по мере

роста проекта монолит становится медленным, хрупким и трудно поддерживаемым.

Микрофронтенды решают эти проблемы ценой усложнения архитектуры. Они вносят дополнительный слой координации (как собирать части, как передавать данные, как синхронизировать версии) и требуют зрелой инфраструктуры (системы сборки, реестры артефактов, мониторинг). Исследования показывают, что переход на микрофронтенды оправдан только при достижении определённого порога сложности проекта и размера команды.

Сравнение с компонентно-ориентированным подходом

Многие разработчики полагают, что микрофронтенды — это просто «компоненты, вынесенные в отдельные репозитории». Однако различие глубже. Компонентный подход (например, использование библиотек UI-компонентов через npm) предполагает, что все компоненты используют общий фреймворк и собираются

в единый бандл. Микрофронтенды же допускают разные фреймворки и развёртывание независимо, а интеграция происходит во время выполнения (runtime).

Таким образом, микрофронтенды обеспечивают более высокий уровень слабой связанности, но ценой потери некоторых статических гарантий (типизация, согласованность стилей).

Сравнение с микросервисами на бэкенде

Микрофронтенды часто называют «микросервисами для фронтенда», но аналогия неполна. На бэкенде микросервисы взаимодействуют синхронно (по HTTP/gRPC) или асинхронно (через очереди), и каждый сервис имеет чёткий API. На клиенте микрофронтенды выполняются в одном браузере, разделяя общую среду (DOM, глобальный объект, память). Это создаёт дополнительные риски конфликтов (глобальные стили, polyfill, версии библиотек) и требует специальных механизмов изоляции.

Кроме того, бэкенд-микросервисы обычно развёртываются на разных серверах, что упрощает физическую изоляцию. Во фронтенде все части загружаются в один контекст, что делает проблему версионности и конкуренции зависимостей более острой.

Основные ограничения и сложности

Производительность. Загрузка нескольких независимых приложений увеличивает число HTTP-запросов и общий объём передаваемого кода, что негативно сказывается на времени загрузки и интерактивности. Особенно критично это для мобильных устройств и медленных сетей. Приходится применять сложные стратегии: динамическая загрузка, предварительная загрузка, сжатие, кэширование.

UI/UX-расходимость. Независимо разрабатываемые микрофронтенды могут использовать разные дизайн-системы, шрифты, анимации, что приводит к несогласованному визуальному опыту. Для решения требуется централизованная система дизайн-токенов, единые компоненты или строгий гайдлайн, что отчасти противоречит принципу автономии.

Литература:

1. Evan, Carter Micro-Frontends: Are They Still Worth It in 2025? / Carter Evan. — Текст: электронный // FeatureSlicedDesign: [сайт]. — URL: <https://feature-sliced.design/ru/blog/micro-frontend-architecture> (дата обращения: 14.07.2026).
2. Stephen, Watts An Introduction to Micro Frontends / Watts Stephen. — Текст: электронный // BMC Blogs: [сайт]. — URL: <https://blogs.bmc.com/micro-frontends/> (дата обращения: 14.07.2026).
3. Madhu, Rebbana Micro Frontend Architecture In Enterprise Applications: Benefits And Challenges / Rebbana Madhu. — Текст: электронный // JournalOf: [сайт]. — URL: <https://jicrcr.com/index.php/jicrcr/article/view/3398> (дата обращения: 14.07.2026).
4. Advantages of applications with micro-frontend architecture. — Текст: электронный // Nauka.ru: [сайт]. — URL: https://naukaru08.ru/en/nauka/conference_article/12332/view (дата обращения: 14.07.2026).

Сложность интеграционного тестирования. Тестирование взаимодействия между микрофронтендами (например, переход с одной страницы на другую с передачей состояния) является нетривиальной задачей. Требуются сквозные тесты, которые работают в окружении, близком к продакшну, и учитывают асинхронную загрузку модулей.

Управление зависимостями. Если разные микрофронтенды используют разные версии одной библиотеки (например, React 17 и React 18), это может привести к конфликтам или дублированию кода. Решения включают изоляцию через iframe, использование Module Federation для дедупликации, или договорённость о единой версии (что снижает гибкость).

Организационные требования

Технические решения — лишь часть успеха. Микрофронтенды требуют зрелой организационной культуры:

- переход от функциональных команд (фронтенд-команда, бэкенд-команда) к кросс-функциональным продуктовым командам, владеющим своими вертикалями;
- внедрение практик DevOps — команды должны уметь самостоятельно развёртывать и эксплуатировать свои микрофронтенды;
- чёткое владение контрактами и API для взаимодействия;
- инвестиции в мониторинг, логирование и отслеживание ошибок на уровне всей системы.

Без этих культурных изменений микрофронтенды приведут к хаосу, а не к улучшению.

Заключение

Микрофронтенды — это мощный, но требовательный инструмент. Сравнение с альтернативами показывает, что он даёт ощутимые преимущества в масштабируемости команд и скорости поставки, но за это приходится платить сложностью инфраструктуры, потенциальными проблемами производительности и необходимостью организационной перестройки. Понимание этих ограничений позволяет принять взвешенное решение о применении микрофронтендов в конкретном проекте.

Динамическая интеграция микрофронтендов: подходы и инструменты

Жарков Никита Александрович, студент
Московский государственный технологический университет «СТАНКИН»

Статья посвящена техническим аспектам интеграции микрофронтендов во время выполнения. Рассматриваются основные подходы: Web Components, Single-SPA и module Federation. Описываются механизмы динамической загрузки модулей, управления маршрутизацией, изоляции стилей и общего состояния. Анализируются достоинства и недостатки каждого подхода, а также приводятся рекомендации по выбору инструментов в зависимости от требований проекта. Особое внимание уделяется вопросам производительности и обновления без перезагрузки страницы.

Введение

В предыдущих статьях мы рассмотрели концепцию микрофронтендов, их преимущества и ограничения. Теперь перейдём к практическим аспектам: как собрать несколько независимых приложений в единый пользовательский интерфейс. В отличие от статической интеграции (когда все части собираются на этапе сборки), микрофронтенды требуют динамической (runtime) композиции, чтобы сохранить возможность независимого развёртывания. Существует несколько подходов, каждый со своими сильными и слабыми сторонами. В данной статье мы дадим обзор основных инструментов и стратегий интеграции.

Основные подходы к интеграции

1. Web Components — Стандартный веб-API, позволяющий создавать кастомные элементы, которые могут быть использованы в любом фреймворке. Каждый микрофронтенд может быть обернут в Web Component и зарегистрирован в браузере. Интеграция происходит просто: добавляется тег `<my-microfrontend>` в HTML. Достоинства: нативная поддержка браузерами, независимость от фреймворков, хорошая изоляция стилей (Shadow DOM).

2. Single-SPA — Библиотека-оркестратор, которая управляет загрузкой и выгрузкой микрофронтендов на основе маршрутизации. Микрофронтенды описываются как «приложения», которые предоставляют функции жизненного цикла (bootstrap, mount, unmount). Single-SPA определяет, какой микрофронтенд активен для текущего URL, и загружает его динамически. Поддерживает разные фреймворки одновременно. Преимущества: хорошая интеграция с системой маршрутов, управление состоянием (через общие шины), лёгкость реализации.

3. Module Federation — Механизм, введённый в Webpack 5 и далее расширенный для Vite, позволяющий загружать модули из разных сборок во время выполнения. Каждый микрофронтенд может «экспортировать» свои модули (компоненты, функции) и «импортировать» модули из других микрофронтендов. Хост-приложение

загружает удалённые манифесты и динамически резолвит зависимости. Это самый современный и мощный подход. Преимущества: минимальные накладные расходы, возможность дедупликации общих зависимостей, поддержка lazy loading на уровне отдельных компонентов, прозрачная интеграция с системой сборки.

Динамическая маршрутизация и управление состоянием

При интеграции микрофронтендов необходимо решить две сквозные задачи:

— **Маршрутизация.** Кто определяет, какой микрофронтенд рендерится при переходе по ссылке? Варианты: единый роутер на уровне хоста (который знает все маршруты) или делегирование роутинга каждому микрофронтенду (каждый управляет своим поддеревом URL). Первый подход проще для координации, второй даёт больше автономии. Single-SPA предлагает гибрид: хост определяет активное приложение по префиксу пути.

— **Общее состояние.** Микрофронтенды часто должны обмениваться данными (например, информация о пользователе, состояние корзины). Для этого используются общие — шины событий (EventBus), глобальный стейт-менеджер (Redux, MobX) через единый экземпляр, или передача данных через URL-параметры и localStorage. Важно избегать жёсткой связанности — лучше использовать события и подписки, чем прямой доступ к состоянию другого микрофронтенда.

Изоляция стилей и предотвращение конфликтов

Одна из главных проблем при интеграции — конфликты CSS. Решения:

- применять Shadow DOM (доступно в Web Components);
- использовать CSS Modules или CSS-in-JS с уникальными именами классов;
- использовать префиксы для селекторов и соглашение о наименовании (BEM);
- применять подход «дизайн-токены» — общие переменные (цвета, отступы) и единый набор компонентов,

но это противоречит принципу автономии, поэтому часто идут на компромисс: базовые токены общие, а кастомизация допустима.

Обновление микрофронтендов без перезагрузки страницы

Одно из ключевых обещаний микрофронтендов — возможность обновлять отдельные части приложения без полной перезагрузки. Это достигается за счёт динамической загрузки новых версий модулей. Module Federation позволяет перезапросить манифест и подгрузить обновлённый код при следующем переходе на соответствующий маршрут. Single-SPA также поддерживает динамическую перерегистрацию приложений. Однако нужно учитывать кеширование браузера и HTTP-заголовки, чтобы пользователи получали актуальные версии.

Производительность: стратегии оптимизации

Динамическая загрузка неизбежно добавляет накладные расходы. Рекомендации:

- минимизировать количество загружаемых микрофронтендов на старте — загружать только то, что видно пользователю;

- использовать предзагрузку (prefetch) и предподготовку (preload) для маршрутов, которые могут быть посещены;

- применять сжатие (gzip/brotli) и кеширование на уровне CDN;

- использовать дедупликацию зависимостей через Module Federation (shared modules).

Заключение

Динамическая интеграция микрофронтендов — это сердцевина подхода, определяющая его реализуемость и эффективность. Выбор инструмента зависит от конкретных требований: Web Components нужен для максимальной стандартизации, Single-SPA — для управляемой маршрутизации, Module Federation — для продвинутой сборки и минимальных накладных расходов. Каждый из подходов активно развивается, и в 2025–2026 годах наблюдается конвергенция: например, Module Federation может использоваться внутри Single-SPA для оптимизации загрузки. Понимание этих механизмов позволяет архитекторам и разработчикам выбирать оптимальную стратегию для своего проекта и эффективно внедрять микрофронтенды, сохраняя автономность команд и высокую производительность.

Литература:

1. An Introduction to Micro Frontends. — Текст: электронный // BMC Blogs. — URL: <https://blogs.bmc.com/micro-frontends/> (дата обращения: 13.07.2026).
2. Fowler, M. Micro Frontends / M. Fowler. — Текст: электронный // martinowler.com. — URL: <https://martinowler.com/articles/micro-frontends.html> (дата обращения: 11.07.2026).
3. Single-SPA Official Documentation. — Текст: электронный // single-spa.js.org. — URL: <https://single-spa.js.org/docs/getting-started-overview> (дата обращения: 13.07.2026).
4. Module Federation Documentation. — Текст: электронный // Webpack. — URL: <https://webpack.js.org/concepts/module-federation/> (дата обращения: 10.07.2026).

Разработка интерактивной виртуальной лаборатории для исследования основной и высшей волны в круглом волноводе

Зарембо Иван Алексеевич, студент магистратуры;

Мительман Юрий Евгеньевич, кандидат технических наук, доцент

Уральский федеральный университет имени первого Президента России Б. Н. Ельцина (г. Екатеринбург)

В статье рассматривается процесс проектирования и разработки интерактивной виртуальной лаборатории для исследования основной и высшей волны в круглом волноводе, выполненной в мультиплатформенной среде разработки Unity.

Ключевые слова: Unity, симуляторы, моделирование, виртуальные лаборатории, учебные компьютерные программы.

В современной системе образования наблюдается постепенное внедрение электронных технологий. Одним из подобных электронных инструментов являются электронные симуляторы, которые могут быть использованы для обучения как студентов медицинских, так и инженерных направлений. К преимуществам использования электронных симуляторов можно отнести возможность их использования при дистанционном обучении, более свободный доступ

студентов к моделируемому оборудованию, исследование аварийных или экстремальных ситуаций без дополнительного риска, снижение потенциальных затрат на приобретение физического оборудования [1].

В данной работе представлен процесс проектирования и разработки одной из трех виртуальных лабораторий по исследованию электромагнитных волн в направляющих системах, разработанных в рамках одного интерактивного приложения-симулятора для студентов, обучающихся на кафедре радиоэлектроники и телекоммуникаций ИРИТ-РТФ. Разработка производилась в мультиплатформенной среде Unity с применением языка программирования C#, т. к. финальный результат должен быть выпущен одновременно в виде настольного приложения для персональных компьютеров и WebGL приложения для его применения на электронной образовательной площадке elearn.

В рамках рассматриваемой лабораторной работы пользователь исследует основную волну H_{11} и высшую волну E_{01} в круглом волноводе. Стенд данной лабораторной работы (рисунок 1) состоит из генератора СВЧ, измерительного усилителя, круглого волновода с вращающейся секцией со штырем для измерения амплитуды радиальной составляющей электрического поля, малый прямоугольный волновод, подсоединенный с помощью кабеля к генератору СВЧ и двух сменяемых переходов от прямоугольного волновода к круглому. Первый переход является плавным и имеет изогнутую в плоскости вектора электрического поля E секцию (данный переход подключен к круглому волноводу на рисунке 1), второй переход является скачкообразным, содержит внутри диэлектрическую втулку для уменьшения поля волны H_{11} и выполняет роль возбудителя двухволнового режима работы установки (данный переход на рисунке 1 отключен от установки и лежит на столе рядом с ней).



Рис. 1. Фотография моделируемой лабораторной установки

Выполнение лабораторной работы поделено на два этапа. На первом этапе при подключенном изогнутом плавном переходе к круглому волноводу, пользователь занимается исследованием волны H_{11} в одноволновом режиме. Для этого вначале, поворачивая вращающуюся секцию определяются одно значения угла поворота, при котором напряженность магнитного поля равно максимуму. При этом зафиксированном положении вращающейся секции, путем перемещения поршня, определяются узлы амплитуды, т. е. значения смещения поршня, при которых напряженность магнитного поля равна нулю. На основе этих значений вычисляется длина волны в волноводе. После путем перемещения поршня с шагом 2–3 мм составляется зависимость напряженности квадрата магнитного поля от положения поршня. После установки поршня в положение, в котором амплитуда электромагнитной волны максимальна, путем поворота вращаю-

щейся секции волновода с шагом 15° снимается зависимость напряженности магнитного поля от положения вращающейся секции. На втором этапе плавный переход к круглому волноводу заменяется на возбудитель двухволнового режима. Вращательная секция устанавливается в положение, в котором значение напряженности магнитного поля основной волны H_{11} равно нулю. Путем перемещения поршня определяют и записывают два положения поршня, при котором значение напряженности магнитного поля высшей волны E_{01} равно нулю. На основе этого вычисляется значение длины волны E_{01} . Аналогично первому этапу работы путем перемещений поршня и вращающейся секции волновода снимаются зависимости значения напряженности магнитного поля от измеренных положений.

На основе описанного алгоритма выполнения лабораторной работы можно выделить отдельные элементы имитационной модели, которые должны быть доступны пользователю для взаимодействия (рисунок 2). Меняя состояния данных компонентов на трехмерных моделях внутри симулятора, пользователь будет менять значения переменных, которые влияют на значение волны, принимаемой измерительным усилителем.



Рис. 2. Диаграмма компонентов имитационной модели лабораторной установки

На основе полученной диаграммы компонентов и теоретических материалов об исследуемых нами физических процессах [2], была построена математическая модель, определяющая значение уровня сигнала на приемнике. Используемые в математическом описании переменные и константы представлены в таблицах 1 и 2. Диапазоны числовых значений были экспериментально взяты на основе лабораторной установки, используемой в ИРИТ-РТФ. Математическое описание волны, распространяемой в круглом волноводе представлено в формулах 1–5.

Длина создаваемой генератором волны в свободном пространстве:

$$\lambda = c/f \quad (1)$$

Длина высшей волны E_{01} в волноводе, м:

$$\lambda_B^{01} = \frac{\lambda}{\sqrt{1 - \left(\frac{\lambda}{2,613a}\right)^2}} \quad (2)$$

Длина основной волны H_{11} в волноводе, м:

$$\lambda_B^{11} = \frac{\lambda}{\sqrt{1 - \left(\frac{\lambda}{3,413a}\right)^2}} \quad (3)$$

z — расстояние от штыря до КЗ поршня вдоль волновода, м,

$$z = z_{КЗ} + z_{ш}$$

$$f(\varphi) = \begin{cases} \sin(\varphi) \cdot \sin\left(\frac{2\pi}{\lambda_{11}^B} z\right) & \text{— для плавного перехода} \\ p_{\text{под}} \cdot \sin(\varphi) \cdot \sin\left(\frac{2\pi}{\lambda_{11}^B} z\right) + 1 \cdot \sin\left(\frac{2\pi}{\lambda_{01}^B} z\right) & \text{— для ступенчатого перехода} \end{cases} \tag{4}$$

Уровень сигнала на приемнике U_s :

$$U_s = p_{\pi} \cdot ((n_{\Gamma} \cdot |f| \cdot p_{\Gamma})^2 \cdot (1 - p_{\text{ун}}) \cdot n_{\pi} + n_{\text{ун}}) \cdot 100 \tag{5}$$

Таблица 1. Переменные, используемые в математической модели

Переменная	Что означает	Принимаемые значения
f	Частота генератора	От 7500 до 10500 ГГц
$z_{\text{кз}}$	Положение поршня	От 0 до 0,07 м
φ	Угол поворота вращательной секции	От 0° до 360°
n_{Γ}	Положение регулятора ослабления на генераторе	От 0 до 2
n_{π}	Положение регулятора усиления на приемнике	От 0 до 2
n_{π}	Положение регулятора установки нуля	От 0 до 1 Начальное положение генерируется случайным образом при запуске в диапазоне [0,04 ... 0,15]
p_{Γ}	Положение выключателя генератора	1 — вкл, 0 — выкл
p_{Γ}	Положение выключателя приемного усилителя	1 — вкл, 0 — выкл
$p_{\text{ун}}$	Положение переключателя установки нуля	1 — вкл, 0 — выкл

Таблица 2. Константы, используемые в математической модели

Константа	Что означает	Значение
a	Радиус волновода	0,015 м
$z_{\text{ш}}$	Расстояние от начала поршня до штыря в круглом волноводе	0,12 м
c	Скорость света	$3 \cdot 10^8$ м/с
$p_{\text{под}}$	Эффективность подавления волны H_{11} в ступенчатом переходе	0,5

В приложении данные формулы, описывающие построенную математическую модель, применяются в программном коде бэкендовой составляющей программы (рисунок 3) из нескольких классов и выводят результат вычислений на дисплей приемника в килогерцах.

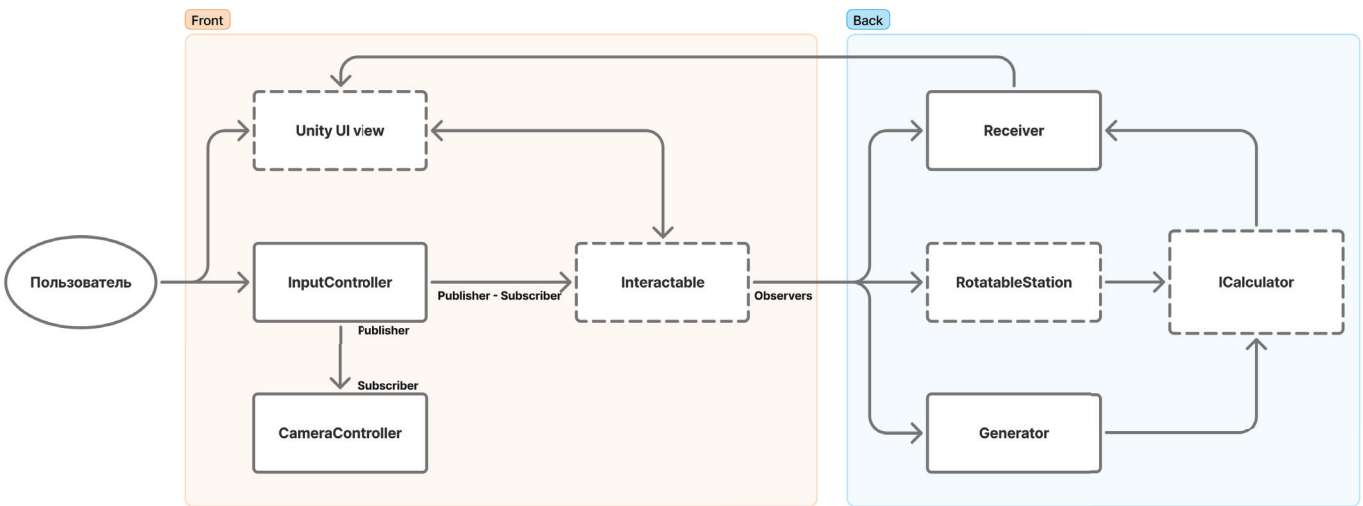


Рис. 3. Архитектура компонентов на игровой сцене в Unity, где представлена модель лабораторной установки

Для настройки зависимых переменных используются трехмерные модели генератора, усилителя и круглого волновода, к интерактивным элементам которых привязаны частные реализации абстрактного класса *Interactable*, которые отвечают за поведение этих компонентов (регуляторов, выключателей, подвижного поршня, вращающейся секции волновода) в зависимости от проводимых пользователем операций. Пользователь взаимодействует с имитационной моделью с помощью компьютерной мыши, перемещение и изменение состояния кнопок которой отслеживается классом *InputController*. Данный класс взаимодействует с реализациями класса *Interactable* и *CameraController*, отвечающий за перемещение камеры и её приближение/отдаление, по паттерну поведения *Publisher — Subscriber*.

Для визуального представления лабораторной установки были созданы трехмерные модели применяемого оборудования с помощью программы *Blender*. Текстурирование готовых трехмерных моделей было выполнено в программе *Adobe Substance 3D Painter* в соответствии стандартом *Universal Render Pipeline* для *Unity*, в результате чего на каждую модель были созданы текстурные карты нормалей(отвечающую за рельефность поверхности), альбедо(отвечающую за базовый цвет поверхности) и металлической гладкости(отвечающую за отражение света поверхностью). Экран готового прототипа представлен на рисунке 4.

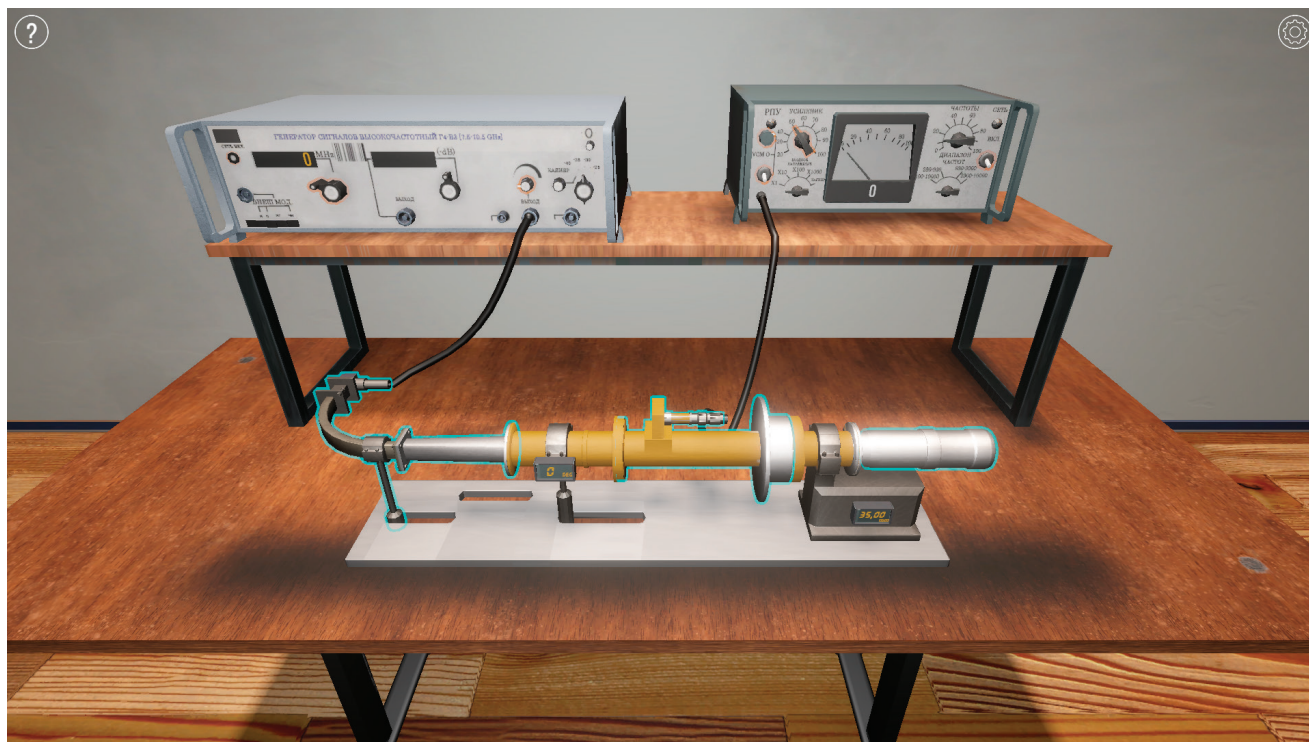


Рис. 4. Экран разработанной виртуальной лаборатории

В ходе данной работы был произведен анализ методических материалов по исследованию электромагнитных волн в направляющих системах, построены диаграммы имитационной модели лабораторной установки и архитектуры сцены в среде *Unity*, построены трехмерные модели реального лабораторного оборудования и разработан прототип приложения для среды *WebGL*. Дальнейшее развитие проекта предполагает создание двух других лабораторных работ, касающихся электромагнитных волн, внутри одного приложения, которое будет передано на кафедру радиоэлектроники и телекоммуникаций ИРИТ-РТФ для обучения студентов.

Литература:

1. Сохатюк, Ю. В. Использование виртуальных лабораторий — фактор повышения качества и эффективности формирования профессиональных компетенций у студентов / Ю. В. Сохатюк. — Текст: непосредственный // Педагогика: традиции и инновации: материалы I Междунар. науч. конф. (г. Челябинск, октябрь 2011 г.). Т. 2. — Челябинск: Два комсомольца, 2011. — С. 146–150.
2. Соловьянова, И. П. Расчет и измерение параметров электромагнитных волн в направляющих системах и на естественных трассах: учебно-методическое пособие / И. П. Соловьянова, Ю. Е. Мительман. — Текст: непосредственный.
3. Козлов, А. В. Цифровизация обучения студентов технических специальностей / А. В. Козлов. — Текст: непосредственный // Современное педагогическое образование. — 2022. — № 10. — С. 201–205.

4. Кирюхина, Н. В. Иммерсивные технологии в обучении физике / Н. В. Кирюхина, Н. А. Плеханова. — Текст: непосредственный // Проблемы современного педагогического образования. — 2023. — № 79. — С. 133–136.
5. Корнеева, Н. Ю. Иммерсивные технологии в современном профессиональном образовании / Н. Ю. Корнеева, Н. В. Уварина. — Текст: непосредственный // Современное педагогическое образование. — 2022. — № 6. — С. 17–22.

Влияние UX-дизайна на эффективность работы пользователей в информационных системах

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Проектирование пользовательского опыта (UX-дизайн) влияет на эффективность работы пользователя не через декоративное оформление интерфейса, а через структуру действий, понятность состояния системы и количество ошибок при выполнении задачи. Цель статьи — рассмотреть, какие элементы проектирования пользовательского опыта помогают пользователю быстрее и точнее работать в информационной системе. В работе анализируются удобство использования, время выполнения задач, частота ошибок, навигация, формы ввода, согласованность интерфейса и измерение пользовательского опыта. Сделан вывод, что проектирование пользовательского опыта повышает эффективность тогда, когда уменьшает лишние решения, делает следующий шаг очевидным и позволяет пользователю исправлять ошибку без потери результата.

Ключевые слова: UX-дизайн, удобство использования, информационная система, эффективность пользователя, ISO 9241–11, SUS, проверка удобства использования, когнитивная нагрузка.

Информационная система должна хранить данные и помогать пользователю выполнять рабочую задачу: найти запись, оформить заявку, проверить статус, внести изменения, сформировать отчет. Если интерфейс заставляет долго искать нужное действие, повторно вводить данные или угадывать причину ошибки, технически корректная система работает хуже для организации. В этом смысле проектирование пользовательского опыта связано с производительностью труда: оно влияет на время выполнения операции, число ошибок, потребность в обучении и нагрузку на поддержку.

В стандарте ISO 9241–11 удобство использования описывается через результативность, эффективность и удовлетворённость в конкретном контексте использования [1]. Для информационной системы это означает, что интерфейс нужно оценивать не абстрактно, а относительно конкретной роли и задачи. Один и тот же экран может быть удобен для опытного оператора, который ежедневно обрабатывает сотни заявок, и неудобен для нового сотрудника, которому нужна подсказка и защита от ошибочного действия. Поэтому проектирование пользовательского опыта начинается с анализа пользователей, сценариев и ограничений рабочей среды.

Эффективность пользователя измеряется через наблюдаемые действия, а не через одно субъективное впечатление. При проверке удобства использования фиксируют долю успешно завершённых задач, время выполнения, число ошибок, число обращений к подсказкам и субъективную оценку сложности. Шкала воспринимаемого удобства использования SUS (System Usability Scale), пред-

ложенная Brooke, даёт короткий опросник для оценки системы [4]. Она не заменяет наблюдение за реальной работой, но помогает сравнить версии интерфейса или выявить проблемные участки после проверки.

Навигация влияет на эффективность потому, что определяет путь пользователя к нужной функции. Если структура меню построена по внутренним модулям разработки, а не по рабочим задачам, пользователь вынужден переводить свою цель на язык системы. Например, сотрудник хочет «продлить договор», а интерфейс требует выбрать между разделами «контрагенты», «документы», «финансы» и «согласования». Хорошая навигация группирует действия по смыслу для пользователя, показывает текущее место и не заставляет помнить скрытые переходы.

Формы ввода — один из главных источников потери времени. Ошибка может появиться из-за неясной подписи поля, неверного формата даты, отсутствия маски, слишком поздней проверки или сообщения вроде «Некорректные данные». Проектирование пользовательского опыта снижает эти ошибки через понятные подписи полей, примеры формата, автозаполнение, сохранение черновика и раннюю проверку. Если система сообщает о проблеме рядом с конкретным полем и объясняет, что нужно исправить, пользователь завершает задачу быстрее. Если ошибка появляется после отправки большой формы, часть времени уже потеряна.

Согласованность интерфейса уменьшает когнитивную нагрузку. Одинаковые действия должны называться одинаково, кнопки подтверждения должны находиться в предсказуемом месте, а опасные операции должны от-

личаться от обычных. Nielsen Norman Group описывает удобство использования как качество, определяющее, насколько легко работать с интерфейсом, и включает обучаемость, эффективность, запоминаемость, ошибки и удовлетворённость [2]. Эти свойства важны для корпоративных систем, где пользователь возвращается к интерфейсу регулярно и ожидает, что изученный способ действия будет работать в соседнем разделе.

Визуальная иерархия помогает пользователю отличать главное от второстепенного. В рабочих системах особенно вредны перегруженные экраны, где таблица, фильтры, действия и предупреждения конкурируют за внимание. Если интерфейс показывает все возможные поля сразу, пользователь тратит время на поиск нужного. Более полезный подход — разделить частые и редкие действия, группировать поля, скрывать дополнительные параметры до необходимости и выделять состояние объекта. При этом «минимализм» не должен удалять важную информацию: оператору склада или специалисту поддержки иногда нужна плотная таблица, но она должна быть сканируемой.

Скорость интерфейса также входит в пользовательский опыт. Если система долго отвечает, пользователь повторно нажимает кнопку, открывает соседние вкладки или не понимает, выполнено ли действие. Индикатор загрузки, блокировка повторной отправки формы и понятный статус операции уменьшают число дублирующих действий. Для пользователя важны фактическая задержка ответа серверной части и обратная связь: он должен видеть, что запрос принят, обрабатывается или завершен ошибкой.

Проектирование пользовательского опыта влияет и на обучение новых пользователей. В информационных системах часто есть регламенты, роли, права доступа и исключения. Если интерфейс отражает рабочий процесс, новичок быстрее понимает последовательность действий. Если экран требует знания внутренних кодов, статусов и сокращений, обучение переносится из интерфейса в инструкции и устные объяснения. Это увеличивает зависимость от опытных сотрудников и повышает риск ошибок при кадровых изменениях.

Метрики пользовательского опыта должны быть связаны с бизнес-процессом. Для системы поддержки это может быть среднее время обработки обращения, число переводов между специалистами и доля повторных обращений. Для учетной системы — время создания документа, число исправлений и доля форм, возвращенных на доработку. Для медицинской или финансовой системы — число критических ошибок и успешность выполнения ре-

гламентных операций. Общая оценка «интерфейс удобен» недостаточна, если не видно, какую рабочую операцию он улучшает.

Отдельная задача проектирования пользовательского опыта — работа с ошибками и граничными состояниями. Пользователь сталкивается с основным сценарием, пустыми списками, недоступными правами, конфликтом версий, обрывом соединения и неверно заполненными полями. Если система объясняет причину проблемы и предлагает следующий шаг, пользователь быстрее возвращается к работе. Например, сообщение «Файл не загружен: превышен размер 20 МБ» полезнее, чем общий текст «Ошибка загрузки». В первом случае человек понимает ограничение и может сразу уменьшить файл или выбрать другой способ передачи.

Для корпоративных информационных систем важна совместимость решений по пользовательскому опыту с регламентом. Интерфейс не должен обходить контрольные точки процесса ради визуальной простоты. Если заявка требует согласования, пользователь должен видеть текущий этап, ответственного и причину задержки. Если действие необратимо, система должна предупредить о последствиях и отделить его от обычных операций. Такой подход уменьшает ошибки не за счет лишних подтверждений, а за счет ясного отображения риска там, где он действительно есть.

Проектирование пользовательского опыта также влияет на работу опытных пользователей. После обучения им нужны быстрые фильтры, горячие действия, сохраненные представления таблиц и предсказуемая навигация. Новичку важны подсказки и защита от ошибок, а опытному сотруднику — минимальное число кликов и возможность быстро обработать поток однотипных задач. Поэтому зрелый интерфейс поддерживает оба режима: объясняет сложные места, но не заставляет каждый раз проходить длинный обучающий путь.

Таким образом, проектирование пользовательского опыта влияет на эффективность работы пользователей через конкретные механизмы: сокращает путь к действию, уменьшает число ошибок, снижает когнитивную нагрузку, ускоряет обучение и делает состояние системы понятным. Его результат следует проверять на реальных задачах и измерять через время, успешность, ошибки и удовлетворенность. Для информационных систем это особенно важно, потому что интерфейс становится частью производственного процесса, а не отдельным визуальным слоем поверх программы.

Литература:

1. ISO. ISO 9241-11:2018 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. — 2018. — URL: <https://www.iso.org/standard/63500.html>.
2. Nielsen Norman Group. Usability 101: Introduction to Usability. — 2012. — URL: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
3. Nielsen J. 10 Usability Heuristics for User Interface Design // Nielsen Norman Group. — 1994. — URL: <https://www.nngroup.com/articles/ten-usability-heuristics/>.

4. Brooke J. SUS: A quick and dirty usability scale. — 1996. — URL: <https://hell.meiert.org/core/pdf/sus.pdf>.
5. Sauro J., Lewis J. R. Quantifying the User Experience: Practical Statistics for User Research. — 2016. — URL: <https://www.sciencedirect.com/book/9780128023082/quantifying-the-user-experience>.
6. Google. The HEART Framework for measuring UX // Google Research. — 2010. — URL: <https://research.google/pubs/measuring-the-user-experience-on-a-large-scale-user-centered-metrics-for-web-applications/>.
7. ISO. ISO 9241–210:2019 Ergonomics of human-system interaction — Human-centred design for interactive systems. — 2019. — URL: <https://www.iso.org/standard/77520.html>.
8. ГОСТ Р ИСО 9241–210–2016. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем. — 2016. — URL: <https://protect.gost.ru/>.

Влияние кэширования на производительность веб-приложений

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Кэширование ускоряет веб-приложение за счет повторного использования уже полученных данных или результатов вычислений. Цель статьи — рассмотреть, как кэширование влияет на время ответа, нагрузку на сервер и удобство работы пользователя. В работе сравниваются HTTP-кэширование, CDN, обратный прокси, прикладной кэш и кэширование запросов к базе данных. Показано, что кэш уменьшает число повторных операций чтения, сериализации, сетевых вызовов и обращений к базе данных, но требует правил инвалидации и контроля устаревших данных. Сделан вывод, что кэширование дает наибольший эффект там, где данные часто читаются и редко меняются.

Ключевые слова: кэширование, производительность веб-приложений, HTTP-кэш, Cache-Control, ETag, CDN, Redis, NGINX, инвалидация кэша.

Производительность веб-приложения складывается из скорости серверного кода, сетевых задержек, числа запросов к базе данных, повторной сериализации одинаковых данных, доставки статических файлов и расстояния между пользователем и сервером. Кэширование уменьшает часть этих затрат: система сохраняет ранее полученный результат и возвращает его без полного повторения исходной операции. Если пользователь второй раз запрашивает тот же CSS-файл, браузеру не нужно скачивать его заново. Если приложение часто читает один и тот же справочник, сервер может взять его из Redis, а не выполнять SQL-запрос.

Первый уровень кэширования находится на стороне браузера и управляется HTTP-заголовками. RFC 9111 описывает механизм HTTP caching и правила, по которым кэш определяет актуальность ответа, необходимость повторной проверки и допустимость повторного использования [1]. Заголовок 'Cache-Control' задает директивы вроде 'max-age', 'no-store', 'private' и 'public'. Валидаторы 'ETag' и 'Last-Modified' позволяют клиенту проверить, изменился ли ресурс, не скачивая его полностью. MDN отдельно показывает, что условный запрос с 'If-None-Match' может привести к ответу '304 Not Modified', если содержимое не изменилось [2]. Для статических ресурсов это сокращает объем передаваемых данных и ускоряет повторные посещения.

Кэш браузера особенно полезен для ресурсов с версионированными именами: JavaScript-бандлов, CSS-файлов, изображений и шрифтов. Если имя файла содержит хэш содержимого, приложение может выставить долгий срок

хранения. Когда файл изменится, изменится и имя, поэтому старый кэш не помешает загрузить новую версию. Если же приложение кэширует файл по стабильному имени, например 'app.js', то после релиза часть пользователей может продолжать получать старую версию. Поэтому сборка фронтенда, HTTP-заголовки и стратегия именования файлов должны проектироваться вместе.

Второй уровень — CDN. CDN размещает копии ресурсов ближе к пользователю и снижает задержку доставки. Для статических файлов эффект очевиден: изображение или скрипт отдается с ближайшего edge-узла. Но CDN может кэшировать и часть динамических ответов, если приложение явно задает правила. Например, публичная страница каталога может храниться несколько минут, тогда как личный кабинет пользователя не должен попадать в общий кэш. Ошибка в директивах 'private', 'public' и 'no-store' может привести к утечке персональных данных, поэтому HTTP-кэширование находится на границе производительности и безопасности.

Обратный прокси, например NGINX, позволяет кэшировать ответы серверного приложения внутри инфраструктуры. Директива 'proxy_cache' сохраняет ответ вышестоящего сервера и выдает его повторно без обращения к приложению [3]. Такой подход полезен для страниц, ответов программного интерфейса и файлов, которые часто запрашиваются многими пользователями. Но прокси-кэш требует строгого правила инвалидации: если данные изменились в базе, кэшированный ответ должен быть удален или признан устаревшим. Иначе пользователь

увидит старую цену, старый статус заказа или устаревшее содержимое страницы.

Прикладной кэш работает внутри серверной логики или рядом с ней. Redis и Memcached часто используют для хранения результатов тяжелых запросов, сессий, справочников, токенов и промежуточных вычислений. Типовой шаблон cache-aside устроен так: приложение сначала ищет значение в кэше, при промахе читает его из базы данных, затем записывает в кэш для следующих запросов. Этот шаблон прост, но в момент массового истечения ключей может возникнуть cache stampede: много запросов одновременно не находят значение и идут в базу. Чтобы снизить этот риск, используют случайный разброс TTL, блокировки на восстановление значения или предварительное обновление горячих ключей.

Кэширование запросов к базе данных требует осторожности. Если приложение кэширует результат SQL-запроса, оно должно понимать, какие таблицы и строки влияют на этот результат. Запрос `SELECT * FROM products WHERE category_id = 5` устареет при изменении любого товара в категории. Чем сложнее запрос и связи между таблицами, тем труднее точно инвалидировать кэш. Поэтому на практике часто кэшируют не произвольные SQL-результаты, а предметные объекты: профиль пользователя, настройки проекта, список доступных ролей, агрегированную статистику за заверченный период.

Влияние кэша на производительность измеряется через долю попаданий в кэш, среднее время ответа, время ответа в хвосте распределения, число запросов к базе данных и нагрузку на серверное приложение. Высокая доля попаданий означает, что значительная часть запросов обслуживается из кэша. Но сам по себе этот показатель не гарантирует пользу. Если кэшируются дешевые ответы, а дорогие операции все равно проходят до базы данных, пользователь может не заметить ускорения. Поэтому полезнее измерять долю попаданий вместе с экономией времени на конкретных маршрутах: сколько миллисекунд занимает запрос при попадании и промахе, сколько обращений к базе данных удалось убрать, как изменились задержки на 95-м и 99-м перцентилях.

Главный риск кэширования — устаревшие данные. Система должна заранее решить, где допустима eventual consistency, а где нужен актуальный ответ. Новостная лента может показывать материал с задержкой в минуту. Баланс счета, статус платежа или список прав доступа должны обновляться строже. Для таких данных кэш либо не используют, либо применяют короткий TTL и инвалидацию после подтвержденной записи. Граница допустимой устарелости должна исходить из предметной области, а не из удобства настройки Redis или CDN.

Кэширование также влияет на отладку. Когда ответ проходит через браузерный кэш, CDN, обратный прокси и Redis, разработчик должен понимать, на каком уровне находится устаревшее значение. Без логирования `cache hit/miss`, метрик TTL и заголовков ответа поиск ошибки превращается в угадывание. Поэтому зрелая стратегия кэширования включает наблюдаемость: заголовки вроде `X-Cache`, метрики попаданий, размер кэша, число evictions, время восстановления значения и предупреждения о росте промахов.

Отдельного внимания требует прогрев кэша после релиза или сбоя. Если популярные ключи удаляются одновременно, первые пользователи после перезапуска получают медленные ответы, а база данных принимает пиковую нагрузку. Для критичных маршрутов применяют warm-up: заранее запрашивают главные страницы, справочники, конфигурации и другие горячие данные. Такой подход не заменяет нормальную инвалидацию, но сглаживает нагрузку и уменьшает риск того, что восстановление сервиса начнется с лавины тяжелых запросов.

Кэш должен иметь понятную политику памяти. Redis, CDN и браузерный кэш не бесконечны: старые значения вытесняются, ключи истекают по TTL, а крупные ответы занимают место быстрее мелких. Если приложение без ограничений кладет в кэш большие JSON-ответы, года может исчезнуть из-за роста памяти, сериализации и сетевого обмена с самим кэшем. Поэтому для каждого уровня нужно определить, какие данные кэшируются, на какой срок, каким ключом и при каком размере ответ лучше не сохранять.

Полезно рассматривать кэширование как часть архитектурного контракта, а не как локальную оптимизацию. Backend должен явно задавать заголовки, CDN — уважать правила приватности, прикладной код — инвалидировать ключи после изменения данных, а мониторинг — показывать промахи и устаревшие ответы. Если хотя бы один уровень работает по неявному правилу, то поведение системы становится трудно объяснить пользователю и разработчику.

Таким образом, кэширование повышает производительность веб-приложений через конкретные механизмы: уменьшает число повторных сетевых передач, обращений к базе данных, сериализаций и вычислений. Наибольший эффект появляется у данных с высокой частотой чтения и понятным правилом обновления. Наименьшая польза — у данных, которые редко повторяются или требуют строгой актуальности. Поэтому кэш следует вводить после анализа маршрутов и метрик, а не как универсальную прослойку перед каждой операцией.

Литература:

1. Fielding R., Nottingham M., Reschke J. RFC 9111: HTTP Caching. — 2022. — URL: <https://www.rfc-editor.org/rfc/rfc9111>.
2. MDN Web Docs. HTTP caching // MDN. — 2026. — URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching> (дата обращения: 05.07.2026).

3. NGINX Documentation. Content Caching // NGINX. — 2026. — URL: <https://docs.nginx.com/nginx/admin-guide/content-cache/content-caching/> (дата обращения: 05.07.2026).
4. Redis Documentation. Caching // Redis. — 2026. — URL: <https://redis.io/solutions/caching/> (дата обращения: 05.07.2026).
5. Amazon Web Services. Caching Best Practices // AWS. — 2026. — URL: <https://aws.amazon.com/caching/best-practices/> (дата обращения: 05.07.2026).
6. Cloudflare Learning Center. What is caching? // Cloudflare. — 2026. — URL: <https://www.cloudflare.com/learning/cdn/what-is-caching/> (дата обращения: 05.07.2026).
7. Google Web.dev. HTTP cache // web.dev. — 2026. — URL: <https://web.dev/articles/http-cache> (дата обращения: 05.07.2026).
8. Yandex Cloud Documentation. CDN // Yandex Cloud. — 2026. — URL: <https://yandex.cloud/ru/docs/cdn/> (дата обращения: 05.07.2026).

Влияние микросервисной архитектуры на сопровождаемость программных систем

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Микросервисная архитектура влияет на сопровождаемость программной системы через границы изменений: часть правок можно изолировать внутри отдельного сервиса, одновременно часть сложности переносится в контракты, сеть, данные и эксплуатацию. Цель работы — определить, при каких условиях разбиение системы на автономные сервисы сокращает область изменения, а при каких увеличивает число согласований между компонентами и командами. В статье анализируются модульность, независимое развёртывание, владение компонентами, тестирование, наблюдаемость, версионирование API и согласованность данных. Основной тезис состоит в том, что микросервисы повышают сопровождаемость только при осознанном выборе границ сервисов и наличии инженерных практик, поддерживающих распределённую систему. Поэтому микросервисы следует оценивать как архитектурный компромисс между локальной изменяемостью и общей сложностью сопровождения.

Ключевые слова: микросервисная архитектура, сопровождаемость, программная инженерия, модульность, DevOps, наблюдаемость, API, согласованность данных, технический долг.

Сопровождаемость программной системы связана со способностью вносить изменения после ввода системы в эксплуатацию: исправлять ошибки, адаптировать функциональность, улучшать характеристики и предотвращать будущие сбои. В терминах качества программного обеспечения это не отдельная фаза, а свойство архитектуры, кода, процессов разработки и эксплуатационной среды. На практике улучшение сопровождаемости можно оценивать по размеру затронутого кода, числу зависимых компонентов, времени от изменения до релиза, частоте неудачных выпусков и среднему времени восстановления после инцидента. Поэтому вопрос о влиянии микросервисной архитектуры нельзя сводить к числу модулей или скорости релизов. Важно понять, где именно уменьшается сложность и где она появляется заново.

Микросервисный подход строится вокруг небольших сервисов, ориентированных на бизнес-возможности и развёртываемых независимо. В формулировке Мартина Фаулера важны размер компонента, автономия команды, изоляция изменений и возможность развивать сервис без одновременного изменения всей системы. Для сопровождаемости это важно за счёт меньшего контекста изменения: команда работает с ограниченным кодом, отдельным циклом релиза, собственным контрактом и понятной зоной эксплуатационного риска.

Однако сама по себе декомпозиция не делает систему сопровождаемой. Если границы сервисов выбраны по техническим слоям, случайным таблицам или текущей организационной структуре, изменение бизнес-правила начинает проходить через несколько сервисов одновременно. В таком случае разработчик получает не меньшую сложность, а распределённую цепочку изменений: нужно менять API, согласовывать версии, обновлять тесты, синхронизировать релизы и отслеживать последствия в нескольких журналах. Поэтому гранулярность сервиса становится одним из ключевых архитектурных решений.

Систематическое исследование перехода к микросервисам показывает, что проблема гранулярности связана с самим процессом миграции. Слишком крупный сервис сохраняет свойства монолита и не даёт локальной изменяемости. Слишком мелкий сервис увеличивает число сетевых вызовов, контрактов и точек отказа. На практике сопровождаемая архитектура возникает между этими крайностями: сервис должен быть достаточно малым, чтобы команда могла владеть им целиком, но достаточно крупным, чтобы изменение доменного правила не превращалось в каскад правок по всей системе.

Положительное влияние микросервисов на сопровождаемость проявляется прежде всего в локализации изменений. Когда сервис имеет чёткий публичный контракт

и собственный цикл развёртывания, исправление дефекта или добавление функции не требует пересборки всей системы. Команда может выпускать небольшие изменения, откатывать неудачный релиз и отслеживать конкретные показатели: время ответа сервиса, число ошибок, частоту отказов, долю изменений, приведших к сбою, и среднее время восстановления. Такая схема особенно полезна в системах, где разные части развиваются с разной скоростью и обслуживаются разными командами.

Эта же автономия создаёт новые требования к инженерной дисциплине. Независимое развёртывание возможно только при стабильных API, совместимых изменениях и управляемом версионировании. Исследования эволюции микросервисных API показывают, что изменение контракта может блокировать потребителей сервиса, если не предусмотрены версии, переходный период и анализ влияния на клиентов. Поэтому сопровождаемость в микросервисной системе зависит от внутреннего качества сервиса и от зрелости управления контрактами между сервисами.

Тестирование также меняет свою роль. В монолитной системе значительная часть ошибок выявляется на уровне модульных и интеграционных тестов внутри одного процесса. В микросервисной архитектуре добавляются сетевые задержки, повторные запросы, частичные отказы, несовпадение версий и асинхронные сценарии. Поэтому тестовый контур должен включать контрактные тесты, интеграционные проверки, тесты устойчивости и автоматизацию в CI/CD. Без этого микросервисы ускоряют выпуск отдельных компонентов, но повышают риск несовместимых изменений.

Наблюдаемость становится обязательным условием сопровождения распределённой системы. Когда пользовательский запрос проходит через несколько сервисов, локальный лог одного приложения уже не показывает всю картину. Практический механизм строится вокруг идентификатора трассировки: он связывает входной запрос, вызовы зависимых сервисов, задержки, ошибки и итоговый ответ пользователю. По такой цепочке команда может определить, в каком сервисе выросло время ответа, где начались повторные запросы и какой компонент вызвал отказ бизнес-сценария. Обзор инструментов трассировки показывает, что такие средства применяются для мониторинга, отладки, поиска узких мест и анализа отказов в микросервисной среде.

Отдельную сложность создаёт работа с данными. Микросервисы часто стремятся к владению собственным хранилищем, чтобы сервис мог развиваться независимо

от внутренней модели других компонентов. Это уменьшает связность на уровне схемы базы данных, но переносит сложность в согласование событий, саги, идемпотентность, повторную доставку сообщений и обработку конфликтов. Например, оформление заказа может затрагивать сервис корзины, оплаты, склада и уведомлений. Если оплата прошла, а резервирование товара завершилось ошибкой, системе нужны компенсирующие действия и понятное правило восстановления состояния. В монолите такая операция чаще укладывается в одну транзакцию, а в микросервисах становится отдельной логикой сопровождения.

Поэтому миграция с монолита на микросервисы должна рассматриваться как управляемое изменение архитектуры, а не как замена одного стиля другим. Отчёты об опыте и исследования случаев показывают, что декомпозиция может улучшать связность, модульность и использование ресурсов, но эффект зависит от выбранного фрагмента системы и зрелости инфраструктуры. Если команда переносит в сервисы тот же технический долг, те же неясные границы ответственности и те же ручные процедуры релиза, сопровождаемость не улучшается, а дефекты становятся труднее локализуемыми.

Организационный аспект не менее важен, чем технический. Микросервисная архитектура хорошо работает там, где команда владеет сервисом от разработки до эксплуатации, понимает его контракт и отвечает за последствия изменений. Если ответственность разделена между аналитиками, разработчиками, тестировщиками и специалистами эксплуатации без общего владения сервисом, то микросервисы увеличивают число согласований. В этом случае архитектурная автономия не превращается в сопровождаемость, потому что решение всё равно проходит через длинную цепочку организационных зависимостей.

Следовательно, влияние микросервисной архитектуры на сопровождаемость программных систем неоднозначно. Она может уменьшить локальную сложность, ускорить выпуск изменений и ограничить область дефекта границами сервиса. Но за это приходится платить распределённой отладкой, управлением API, тестированием сетевых сценариев, трассировкой, консистентностью данных и координацией команд. Микросервисы повышают сопровождаемость только тогда, когда границы сервисов соответствуют доменной модели, изменения проходят через автоматизированный контур проверки, а эксплуатационные инструменты показывают цепочку вызовов, ошибки по сервисам, задержки и зависимые запросы.

Литература:

1. ГОСТ Р ИСО/МЭК 14764–2002. Информационная технология. Сопровождение программных средств. — М.: Госстандарт России, 2002.
2. ГОСТ Р ИСО/МЭК 25010–2015. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов. — М.: Стандартинформ, 2015.

3. Орлик С., Булуй Ю. Программная инженерия и управление жизненным циклом // Software-Testing.Ru. — 2008. — URL: <https://software-testing.ru/library/around-testing/engineering/267-swebok> (дата обращения: 05.07.2026).
4. Fowler M. Microservices. — 2014. — URL: <https://martinfowler.com/articles/microservices.html>.
5. Balalaie A., Heydarnoori A., Jamshidi P. Migrating to Cloud-Native Architectures Using Microservices: An Experience Report. — 2015. — URL: <https://arxiv.org/abs/1507.08217>.
6. Amaral M., Polo J., Carrera D., Mohamed I., Unuvar M., Steinder M. Performance Evaluation of Microservices Architectures using Containers. — 2015. — URL: <https://arxiv.org/abs/1511.02043>.
7. Shahin M., Babar M. A., Zhu L. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. — 2017. — URL: <https://arxiv.org/abs/1703.07019>.
8. Hassan S., Bahsoon R., Kazman R. Microservice Transition and its Granularity Problem: A Systematic Mapping Study. — 2019. — URL: <https://arxiv.org/abs/1903.11665>.
9. Taibi D., Lenarduzzi V., Pahl C. Continuous Architecting with Microservices and DevOps: A Systematic Mapping Study. — 2019. — URL: <https://arxiv.org/abs/1908.10337>.
10. Waseem M., Liang P., Shahin M. A Systematic Mapping Study on Microservices Architecture in DevOps. — 2020. — URL: <https://arxiv.org/abs/2008.07729>.
11. Waseem M., Liang P., Shahin M., Di Salle A., Márquez G. Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective. — 2021. — URL: <https://arxiv.org/abs/2108.03384>.
12. Vale G., Wiese I., Márquez G., Astudillo H., Kalinowski M., Bastos R., Fernández D. M. Designing Microservice Systems Using Patterns: An Empirical Study on Quality Trade-Offs. — 2022. — URL: <https://arxiv.org/abs/2201.03598>.
13. Barzotto T. R. H., Farias K. Evaluation of the Impacts of Decomposing a Monolithic Application into Microservices: A Case Study. — 2022. — URL: <https://arxiv.org/abs/2203.13878>.
14. Janes A., Li X., Lenarduzzi V. Open Tracing Tools: Overview and Critical Comparison. — 2022. — URL: <https://arxiv.org/abs/2207.06875>.
15. Pereira P., Silva A. R. Towards Transactional Causal Consistent Microservices Business Logic. — 2022. — URL: <https://arxiv.org/abs/2212.11658>.
16. Lercher A., Haupt F., Di Francesco P., Wurster M., Leymann F. Microservice API Evolution in Practice: A Study on Strategies and Challenges. — 2023. — URL: <https://arxiv.org/abs/2311.08175>.

Использование графовых баз данных для анализа связанной информации

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Графовые базы данных применяются там, где ценность информации определяется не только свойствами отдельных объектов, но и связями между ними. Цель статьи — рассмотреть, как графовая модель помогает анализировать связанную информацию в задачах поиска зависимостей, выявления сообществ, построения рекомендательных систем, анализа знаний и расследования сложных цепочек событий. Показано, что графовые базы данных не заменяют все виды хранения, но дают преимущество при частых обходах связей, гибком добавлении новых типов отношений и наглядном представлении структуры предметной области.

Ключевые слова: графовые базы данных, связанные данные, вершины, рёбра, графовая модель, RDF, SPARQL, граф знаний, анализ данных.

Связанная информация плохо описывается плоской таблицей, когда основной вопрос касается не отдельной записи, а пути между объектами. В социальной сети важны не только пользователи, но и отношения подписки, переписки и совместного участия в группах. В информационной безопасности важны связи между учётными записями, устройствами, правами доступа, событиями входа и сетевыми адресами. В научной базе важны авторы, публикации, организации, темы, цитирования и проекты. Графовая база данных хранит такие элементы как вершины и рёбра, поэтому связь становится

частью модели, а не побочным результатом соединения таблиц.

Вершина в графовой базе обычно представляет объект: человека, документ, товар, сервер, организацию или понятие. Ребро описывает отношение между объектами: «купил», «цитирует», «работает в», «зависит от», «подключён к», «похож на». У вершины и ребра могут быть свойства: дата, вес, тип, источник, уровень доверия, описание события. Благодаря этому аналитик может задавать вопросы о структуре связей: кто находится между двумя объектами, какие элементы образуют плотную группу,

где проходит кратчайший путь, какие вершины имеют большое число входящих связей и какие отношения повторяются чаще всего.

Реляционная база данных тоже может хранить связи через внешние ключи и таблицы соответствия. Такой подход хорошо работает для устойчивых бизнес-процессов, где структура данных заранее известна и запросы в основном выбирают строки по условиям. Но при глубоком обходе связей запрос быстро усложняется: нужно многократно соединять таблицы, а число соединений растёт вместе с длиной пути. Графовая база данных проектируется с учётом таких обходов. Она позволяет выразить вопрос ближе к предметной области: найти цепочку, соседей, общих участников, зависимые объекты или подграф вокруг выбранной вершины.

Существуют две распространённые группы графовых моделей. Первая — граф свойств, где вершины и рёбра имеют метки и наборы свойств. Эта модель удобна для прикладных систем, потому что легко описывает объекты с разными характеристиками и отношения с собственными атрибутами. Вторая — модель RDF, то есть Resource Description Framework, стандарт описания ресурсов через тройки «субъект — предикат — объект». RDF часто применяют для семантических данных и графов знаний, где важны единые словари, онтологии и обмен данными между организациями.

Для графов свойств используются языки запросов, ориентированные на образец связей. Запрос описывает, какие вершины и рёбра нужно найти и какие условия должны выполняться для их свойств. В таких языках удобно формулировать задачи вида «найти пользователей, которые покупали похожие товары», «найти сервисы, зависящие от выбранной базы данных», «найти сотрудников, связанных через общие проекты». Для RDF широко применяется SPARQL — язык запросов к данным, представленным в виде троек. Он позволяет искать шаблоны троек, фильтровать результаты и объединять сведения из разных источников.

Графовый анализ начинается с правильной постановки вопроса. Если нужно найти отдельный объект по атрибуту, графовая база не даёт обязательного преимущества перед реляционной таблицей или документным хранилищем. Если нужно пройти по отношениям, сравнить окружения объектов или оценить положение вершины в сети, графовая модель становится полезной. Например, при анализе мошенничества важны цепочки переводов, общие устройства, повторяющиеся номера телефонов и адреса доставки. Каждая отдельная связь может выглядеть законной, но их совокупность показывает подозрительную группу.

В рекомендательных системах граф помогает учитывать связи разных типов. Пользователь связан с товарами через покупки и просмотры, товары связаны между собой категориями, брендами, совместными покупками и отзывами. Графовый подход позволяет искать похожие товары не только по текстовому описанию, но и по поведению пользователей и структуре взаимодействий. Если два пользова-

теля не совпадают по явным признакам, но имеют похожие маршруты в графе, система может предложить новые объекты с более точным объяснением: рекомендация основана на общих покупках, интересах или связанной тематике.

В графах знаний вершины представляют понятия, сущности и события, а рёбра задают смысловые отношения между ними. Такой подход помогает объединять данные из разных источников, потому что новая информация добавляется как новые тройки или новые свойства, не требуя полного изменения схемы. Граф знаний полезен в поисковых системах, справочных системах, научных каталогах и корпоративных хранилищах знаний. Он позволяет отвечать на вопросы не только по совпадению слов, но и по связям: какие организации участвовали в проекте, какие публикации относятся к теме, какие методы применялись в похожих задачах.

Графовые базы данных применяются и в управлении программными системами. Компоненты приложения, библиотеки, интерфейсы, серверы, контейнеры и права доступа образуют сеть зависимостей. Если один компонент меняется, важно понять, какие сервисы он затронет и какие цепочки вызовов могут нарушиться. Графовая модель позволяет хранить карту зависимостей и быстро находить область влияния изменения. Такой анализ помогает при сопровождении больших систем, оценке технического долга и расследовании отказов.

В информационной безопасности графовая база помогает связывать события, которые в журналах выглядят разрозненно. Учётная запись, рабочая станция, сетевой адрес, процесс, файл и запрос к серверу становятся вершинами, а события доступа — рёбрами с временем и типом действия. По такому графу можно искать необычные маршруты: вход с нового устройства, переход к привилегированному ресурсу, запуск редкого процесса, обращение к критическому серверу. Граф не отменяет сигнатурные и статистические методы, но даёт удобный слой для расследования связей между событиями.

Качество графового анализа зависит от качества исходных данных. Если одно и то же лицо записано под разными именами, граф распадается на несколько вершин, и часть связей теряется. Если разные сущности случайно объединены в одну вершину, анализ показывает ложные маршруты. Поэтому перед загрузкой данных нужны очистка, сопоставление сущностей, нормализация идентификаторов и фиксация источника каждого факта. Для графа знаний также важны словари и правила именования отношений: без них разные команды будут описывать одну связь разными способами.

Проектирование графовой схемы требует баланса между гибкостью и управляемостью. Слишком общая схема, где все рёбра называются «связано с», не даёт аналитической пользы. Слишком подробная схема создаёт множество редких типов отношений, которые сложно поддерживать и объяснять пользователям. Хорошая модель отражает реальные вопросы: какие пути нужно искать, какие связи сравнивать, какие свойства фильтровать,

какие результаты должны быть проверяемыми. Поэтому графовую модель лучше строить не от абстрактной онтологии, а от набора аналитических сценариев.

Производительность графовой базы определяется не только объёмом данных, но и характером запросов. Поиск ближайших соседей и коротких путей обычно выполняется быстро при правильных индексах и локальности данных. Глубокий обход с большим числом вариантов может стать дорогим, потому что количество возможных путей растёт очень быстро. Поэтому запросы нужно ограничивать длиной пути, типами рёбер, временным окном и условиями на свойства. Для тяжёлой аналитики иногда применяют отдельные графовые вычислительные движки, которые заранее рассчитывают центральность, сообщество или другие показатели.

Графовые базы данных не всегда подходят как основное хранилище системы. Транзакционные операции с простыми таблицами, строгая финансовая отчётность или массовая обработка событий могут эффективнее выполняться в реляционных или потоковых системах. На практике графовую базу часто используют как аналитический слой рядом с другими хранилищами. Данные поступают из журналов, таблиц, документов и внешних справочников, затем приводятся к единой графовой модели. Такой подход снижает риск замены всей информационной системы и позволяет применять графовый анализ там, где он действительно нужен.

Для принятия решения о внедрении графовой базы данных полезно проверить несколько критериев. В предметной области должно быть много отношений между объектами, а типовые вопросы должны требовать обхода этих отношений. Пользователям нужны не только списки объектов, но и объяснимые цепочки связей. Схема должна меняться чаще, чем в классической табличной модели, но при этом основные типы сущностей и отношений должны быть понятны. Если эти условия выполняются, графовая база может сократить сложность запросов и сделать анализ ближе к естественному описанию задачи.

Следовательно, графовые базы данных полезны не потому, что они новее реляционных систем, а потому, что явно представляют отношения как данные первого уровня. Это меняет способ анализа: исследователь работает не только с наборами записей, но и с маршрутом, окружением, плотностью связей, центральными вершинами и подграфами. При грамотной подготовке данных графовая модель помогает находить скрытые зависимости, объяснять рекомендации, расследовать инциденты и объединять знания из разных источников. Ограничения тоже существенны: плохие идентификаторы, хаотичные типы отношений и неограниченные обходы быстро снижают пользу. Поэтому графовая база данных должна внедряться как инструмент для конкретных аналитических вопросов, а не как универсальная замена всем хранилищам.

Литература:

1. Angles R., Gutierrez C. Survey of graph database models. — ACM Computing Surveys. — 2008. — DOI: 10.1145/1322432.1322433. — URL: <https://dl.acm.org/doi/10.1145/1322432.1322433>.
2. Angles R., Arenas M., Barceló P., Boncz P., Fletcher G., Gutierrez C., Lindaaker T., Paradies M., Plantikow S., Sequeda J., van Rest O., Voigt H. G-CORE: A Core for Future Graph Query Languages. — 2018. — URL: <https://arxiv.org/abs/1712.01550>.
3. Hogan A., Blomqvist E., Cochez M., d'Amato C., de Melo G., Gutierrez C., Kirrane S., Gayo J. E. L., Navigli R., Neumaier S., Ngomo A.-C. N., Polleres A., Rashid S. M., Rula A., Schmelzeisen L., Sequeda J., Staab S., Zimmermann A. Knowledge Graphs. — 2021. — URL: <https://arxiv.org/abs/2003.02320>.
4. W3C. RDF 1.1 Concepts and Abstract Syntax. — 2014. — URL: <https://www.w3.org/TR/rdf11-concepts/>.
5. W3C. SPARQL 1.1 Query Language. — 2013. — URL: <https://www.w3.org/TR/sparql11-query/>.
6. Apache Software Foundation. Apache TinkerPop Documentation. — URL: <https://tinkerpop.apache.org/docs/current/reference/>.

Использование компьютерного зрения для распознавания объектов в реальном времени

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Компьютерное зрение для распознавания объектов в реальном времени используется там, где система должна обнаружить объект на видеопотоке и быстро передать результат следующему компоненту. Цель статьи — рассмотреть, какие методы и ограничения определяют практическое применение обнаружения объектов в реальном времени. В работе сравниваются двухэтапные и одноэтапные детекторы, YOLO (You Only Look Once), SSD (Single Shot MultiBox

Detector), Faster R-CNN (Faster Region-based Convolutional Neural Network), качество набора данных, задержка, аппаратное ускорение и развертывание на периферийных устройствах. Сделан вывод, что распознавание в реальном времени требует баланса между точностью, частотой кадров, размером модели и ценой ложного срабатывания.

Ключевые слова: компьютерное зрение, распознавание объектов в реальном времени, YOLO, SSD, Faster R-CNN, OpenCV, TensorRT, периферийный ИИ.

Распознавание объектов в реальном времени отличается от обычной обработки изображений тем, что результат нужен быстро и непрерывно. Система получает кадры с камеры, выделяет объекты, определяет их классы и передает координаты дальше: в интерфейс оператора, робототехнический контроллер, систему безопасности или аналитический модуль. Если обработка одного кадра занимает слишком много времени, решение устаревает: объект уже переместился, а система реагирует на прошлое состояние сцены.

Классическая задача обнаружения объектов включает два результата: класс объекта и его положение на изображении. В отличие от классификации, где модель отвечает, что изображено на всей картинке, детектор должен найти несколько объектов и построить ограничивающие рамки. Для оценки качества используют точность положительных срабатываний (precision), полноту обнаружения (recall), IoU (Intersection over Union — пересечение над объединением) и mAP (mean Average Precision — средняя точность по классам и порогам). Для сценария реального времени к ним добавляется задержка: сколько миллисекунд проходит от получения кадра до результата. Иногда модель с высокой mAP непригодна, если она не успевает обрабатывать поток с нужной частотой.

Двухэтапные детекторы сначала предлагают области-кандидаты, а затем классифицируют их. Faster R-CNN стал важным шагом в этом направлении: RPN (Region Proposal Network — сеть предложения областей) позволила объединить генерацию регионов и классификацию в одну обучаемую архитектуру [3]. Такие методы часто дают высокую точность, но требуют больше вычислений. Они подходят для задач, где важна детальность анализа и допустима большая задержка, например для предварительной разметки изображений или аналитики с невысокой частотой кадров.

Одноэтапные детекторы решают задачу быстрее: они предсказывают классы и координаты объектов за один проход модели. YOLO сформулировал обнаружение объектов как единую задачу регрессии и был ориентирован на работу в реальном времени [1]. SSD также относится к одноэтапным методам и использует набор опорных рамок разных масштабов для обнаружения объектов [2]. Эти подходы стали основой многих практических систем, потому что дают приемлемый компромисс между точностью и скоростью.

Выбор модели зависит от сценария. Для видеонаблюдения на сервере можно использовать более тяжелый детектор, если потоков немного и есть GPU (graphics processing unit — графический процессор). Для камеры на

производственной линии важна стабильная задержка: система должна реагировать до того, как объект покинет рабочую зону. Для мобильного или периферийного устройства важны размер модели, энергопотребление и тепловой режим. Поэтому вопрос «какая модель лучше» некорректен без ограничений по FPS (frames per second — кадры в секунду), разрешению, числу классов и цене ошибки.

Качество данных часто влияет на результат сильнее, чем выбор архитектуры. Если обучающий набор содержит только хорошо освещенные изображения, модель может ошибаться ночью, при бликах, частичном перекрытии или нестандартном угле камеры. Набор COCO (Common Objects in Context — распространенные объекты в контексте) стал одним из базовых наборов данных для обнаружения объектов и содержит изображения с объектами в естественных контекстах [4]. Но промышленная система обычно требует собственного набора данных: камеры, фон, расстояние до объекта и частота редких случаев отличаются от публичного набора.

Реальное время требует оптимизации всего конвейера обработки кадра. Задержка складывается из захвата кадра, декодирования, изменения размера, нормализации, вывода модели, постобработки, передачи результата и отображения. Нейросеть может работать за 10 мс, но вся система выдавать результат за 80 мс из-за копирования данных между CPU (central processing unit — центральный процессор) и GPU или медленной постобработки. Поэтому в промышленной эксплуатации измеряют сквозную задержку, а не время одного вызова `model.forward`.

Аппаратное ускорение меняет границы применимости. GPU позволяет параллельно выполнять операции свертки, а специализированные библиотеки оптимизируют граф модели. NVIDIA TensorRT предназначен для оптимизации вывода модели и может применять преобразования графа, калибровку точности вычислений и автоматический подбор ядер [7]. OpenCV DNN (Deep Neural Network — глубокая нейронная сеть), ONNX Runtime (Open Neural Network Exchange Runtime — среда выполнения открытого формата нейросетей) и другие инструменты позволяют запускать модели в разных средах [5]. Но ускорение требует проверки точности: квантизация до INT8 (8-bit integer — 8-битное целое число) может уменьшить задержку, но иногда снижает качество на малых объектах или редких классах.

Ошибки распознавания имеют разную цену. В системе подсчета посетителей ложное срабатывание может исказить статистику. В системе безопасности пропуск опасного объекта уже создает риск. В робототехнике неверная

координата может привести к неправильному движению. Поэтому порог уверенности нельзя выбирать только по общей mAP. Его нужно настраивать по сценариям: где важнее не пропустить объект, а где важнее не создавать лишние тревоги. Иногда система должна показывать оператору класс объекта вместе с уверенностью модели.

Внедрение систем компьютерного зрения реального времени включает наблюдаемость и обновление модели. После запуска нужно отслеживать FPS, задержку, распределение уверенности, долю ручных подтверждений, ошибки на конкретных камерах и дрейф данных. Если меняется освещение, фон или тип объектов, качество модели может снизиться без изменения кода. Поэтому практическая система должна сохранять примеры ошибок, поддерживать дообучение и иметь процедуру безопасного обновления модели.

Важную роль играет предобработка видеопотока. Камера может давать шум, размытые кадры, пересвет, сжатие с артефактами и нестабильную частоту кадров. Простые операции вроде нормализации размера, коррекции цвета, фильтрации слишком темных кадров или выбора области интереса могут уменьшить нагрузку на модель и повысить устойчивость результата. При этом чрезмерная предобработка тоже добавляет задержку, поэтому ее оценивают вместе с выводом модели, а не отдельно от основного конвейера.

Для реального времени важна синхронизация результата с исходным кадром. Если система отображает огра-

нивающие рамки поверх видеопотока, задержка между кадром и разметкой должна быть предсказуемой. При перегрузке очередь кадров может расти, и оператор увидит аккуратное, но устаревшее обнаружение. В таких сценариях лучше пропускать часть кадров или обрабатывать последний доступный кадр, чем сохранять полный буфер и увеличивать задержку. Это особенно важно для робототехники, контроля доступа и производственных линий.

Инженерная проверка модели должна включать не только общий набор тестовых изображений, но и сценарии отказа. Нужно проверять частичное перекрытие объектов, быстрое движение, слабое освещение, похожие классы, загрязнение камеры и редкие ракурсы. Такой набор помогает заранее увидеть, где модель дает ложные срабатывания или пропускает объект. После внедрения эти случаи становятся основой для доработки данных и следующего цикла обучения.

Таким образом, компьютерное зрение для распознавания объектов в реальном времени строится на компромиссе между точностью и задержкой. YOLO и SSD полезны там, где нужен быстрый результат, Faster R-CNN — там, где допустима большая вычислительная стоимость ради точности. Но архитектура модели — только часть решения. Для устойчивой работы нужны подходящий набор данных, измерение сквозной задержки, аппаратная оптимизация, настройка порогов и контроль ошибок после внедрения.

Литература:

1. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. — 2016. — DOI: 10.48550/arXiv.1506.02640. — URL: <https://arxiv.org/abs/1506.02640>.
2. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. SSD: Single Shot MultiBox Detector. — 2016. — DOI: 10.48550/arXiv.1512.02325. — URL: <https://arxiv.org/abs/1512.02325>.
3. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. — 2015. — DOI: 10.48550/arXiv.1506.01497. — URL: <https://arxiv.org/abs/1506.01497>.
4. Lin T.-Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C. L. Microsoft COCO: Common Objects in Context. — 2014. — DOI: 10.48550/arXiv.1405.0312. — URL: <https://arxiv.org/abs/1405.0312>.
5. OpenCV Documentation. Deep Neural Networks module // OpenCV. — 2026. — URL: https://docs.opencv.org/4.x/d2/d58/tutorial_table_of_content_dnn.html (дата обращения: 05.07.2026).
6. Ultralytics Documentation. YOLO object detection // Ultralytics. — 2026. — URL: <https://docs.ultralytics.com/tasks/detect/> (дата обращения: 05.07.2026).
7. NVIDIA Documentation. NVIDIA TensorRT Documentation // NVIDIA. — 2026. — URL: <https://docs.nvidia.com/deeplearning/tensorrt/> (дата обращения: 05.07.2026).
8. ONNX Runtime Documentation. Performance and model optimizations // ONNX Runtime. — 2026. — URL: <https://onnxruntime.ai/docs/performance/> (дата обращения: 05.07.2026).

Прогностические модели на основе машинного обучения как инструмент оптимизации закупочной деятельности в условиях волатильности сырьевых рынков

Лихинин Александр Евгеньевич, студент магистратуры
Московский государственный технологический университет «СТАНКИН»

Исследование направлено на обоснование применения прогностических моделей машинного обучения для оптимизации закупочной деятельности промышленных предприятий в условиях высокой волатильности сырьевых рынков. На основе анализа современных подходов к прогнозированию временных рядов (LSTM, Transformer, XGBoost, гибридные архитектуры) и методов оптимизации решений (обучение с подкреплением, динамическое программирование) предложена концептуальная схема интеллектуальной системы управления закупками, интегрирующая блоки прогнозирования цен и спроса с блоком оптимизации решений. Показано, что современные модели глубокого обучения, учитывающие макроэкономические индикаторы и новостной фон, обеспечивают снижение ошибки прогнозирования на 15–30 % по сравнению с классическими статистическими методами; применение прогнозно-оптимизационных алгоритмов позволяет сократить закупочные затраты на 4–14,5 % и снизить число дефицитных событий до 68,8 %. Прогностические модели на основе машинного обучения являются эффективным инструментом повышения обоснованности закупочных решений; их практическая реализация требует решения проблем качества данных, интерпретируемости и интеграции с существующими системами управления предприятием.

Ключевые слова: прогнозирование цен на сырье, машинное обучение, оптимизация закупок, управление цепями поставок, волатильность рынков.

Волатильность сырьевых рынков достигла уровня, при котором традиционные подходы к планированию закупок теряют эффективность. Геополитические конфликты, нарушения цепочек поставок и климатические факторы создают среду, где цены на критически важное сырье могут изменяться на десятки процентов за короткий период [1]. Структурные сдвиги в мировой экономике ведут к сырьевой биполяризации, когда разные группы стран оказываются в различных ценовых зонах [5]. По оценкам ЮНКТАД, нестабильность на сырьевых рынках носит системный характер и требует пересмотра подходов к управлению закупочными рисками [4]. В этих условиях ключевым направлением становится внедрение прогностических моделей на основе машинного обучения. Стратегическое планирование закупок должно обеспечивать не только ценовую эффективность, но и бесперебойность поставок, что в равной степени зависит от качества прогнозов [6].

Традиционные методы прогнозирования (ARIMA, экспоненциальное сглаживание) предполагают линейность и стационарность временных рядов, что не соответствует реальной динамике рынков. Машинное обучение позволяет выявлять сложные нелинейные зависимости и учитывать широкий спектр внешних факторов. Цель работы — систематизировать современные подходы к прогнозированию цен на сырье и обосновать их прикладное значение для оптимизации закупок.

Современные исследования предлагают широкий спектр моделей для прогнозирования сырьевых цен. Рекуррентные сети, в частности LSTM, демонстрируют способность улавливать долгосрочные зависимости в нелинейных данных, превосходя традиционные статисти-

ческие методы. Модель Trend-Calibrated Autoformer (TCA), основанная на механизме автокорреляционного кодирования и динамической калибровки тренда, показывает превосходство в многошаговом прогнозировании. Анализ факторов с помощью TCA демонстрирует, что драйверы волатильности различаются в зависимости от горизонта прогнозирования: краткосрочные колебания определяются новостным фоном, среднесрочные — финансовыми индикаторами, долгосрочные — макроэкономическими условиями.

Важным направлением является интеграция разнородных данных. Включение макроэкономических показателей (инфляция, валютные курсы) значительно повышает точность прогнозов. Применение методов обработки естественного языка для извлечения сигналов настроений из новостных и социальных медиа позволяет учитывать поведенческий фактор [8]. Гибридные подходы, объединяющие глубокое обучение для числовых рядов и NLP для текстов, становятся новой парадигмой в прогнозировании сырьевых цен.

Прогнозирование цен само по себе не решает задачу оптимизации закупок. Необходим переход к системе поддержки принятия решений, определяющей объемы и сроки закупок. В этом контексте перспективным является обучение с подкреплением (Reinforcement Learning), позволяющее адаптивно выбирать стратегию в зависимости от состояния рынка. В условиях усиливающихся «медвежьих» рисков [7] такие алгоритмы приобретают особую ценность, позволяя оперативно перестраивать стратегию при смене конъюнктуры.

Метод динамического обучения с подкреплением с механизмом воспроизведения опыта (DRRL) использует

LSTM для краткосрочного прогнозирования цен и спроса, затем принимает последовательные решения об объемах закупок. Другой подход сочетает прогностический модуль на основе STL-Transformer с агентом РРО, что позволяет сократить затраты и снизить число дефицитных событий. Наиболее эффективные системы строятся по принципу замкнутого контура: «прогнозирование — принятие решения — калибровка». Результаты закупочных решений анализируются для корректировки моделей. Ключевым вызовом остается не разработка моделей, а организация устойчивого потока релевантных данных и их качественная предобработка [6].

Эмпирические исследования демонстрируют значимые экономические эффекты. Сравнение LSTM-стратегии с традиционной системой закупок показало сокращение затрат на 4 %. Комплексная система, объединяющая прогнозирование и оптимизацию с помощью обучения с подкреплением, позволила снизить затраты на 14,5 %. В российском контексте, по данным разработчиков специализированных платформ, применение прогностического моделирования в нефтехимии позволяет значительно увеличить маржинальность бизнеса.

Качество и полнота исходных данных — важнейший фактор успеха. Необходим автоматизированный сбор внутренних данных (запасы, производственные планы, исторические цены) и внешних макроэкономических показателей. Внедрение требует создания устойчивого конвейера данных и регулярного обновления моделей. Как

показывает анализ рыночных тенденций, волатильность будет сохраняться в обозримом будущем [2] [3], что делает инвестиции в прогностические инструменты стратегически необходимыми. Сырьевой фактор остается ключевым источником волатильности на глобальных рынках [3], а перспективы товарно-сырьевых рынков указывают на сохранение нестабильности [2].

Прогностические модели на основе машинного обучения, особенно архитектуры глубокого обучения (LSTM, Transformer), существенно превосходят традиционные методы по точности прогнозирования цен на сырье в условиях высокой волатильности. Наибольший практический эффект достигается при интеграции прогнозного модуля с алгоритмами оптимизации решений (обучение с подкреплением), что позволяет адаптивно определять объемы и сроки закупок. Экономическая эффективность подтверждается эмпирически: сокращение затрат составляет 4–14,5 %, число дефицитных событий снижается до 68,8 %.

Сдерживающими факторами остаются проблемы качества данных, интерпретируемости моделей и интеграции с существующими системами управления. Дальнейшие исследования должны быть направлены на разработку интерпретируемых архитектур и адаптацию моделей к отраслевой специфике. Учитывая сохраняющуюся нестабильность сырьевых рынков [4], развитие таких систем становится необходимым условием устойчивости бизнеса.

Литература:

1. Forbes.ru. Эра спекулянтов: чем грозит России рекордная волатильность сырьевых товаров [Электронный ресурс]. — 2025. — URL: <https://www.forbes.ru/biznes/541981-era-spekulantov-chem-grozit-rossii-rekordnaa-volatilnost-syr-evyh-tovarov> (дата обращения: 14.07.2026).
2. Roscongress. Перспективы товарно-сырьевых рынков [Электронный ресурс]. — 2024. — URL: <https://roscongress.org/materials/perspektivy-tovarno-syrevykh-rynkov/> (дата обращения: 14.07.2026).
3. Finam. Сырьевой фактор остается ключевым источником волатильности [Электронный ресурс]. — 2026. — URL: <https://www.finam.ru/publications/item/syrevoy-faktor-ostaetsya-klyuchevym-istochnikom-volatilnosti-20260310-0900/> (дата обращения: 14.07.2026).
4. United Nations Conference on Trade and Development. TD/B/C.I/MEM.2/30 [Электронный ресурс]. — URL: https://unctad.org/system/files/official-document/cimem2d30_ru.pdf (дата обращения: 14.07.2026).
5. ECONS.ONLINE. Сырьевая биполяризация мира [Электронный ресурс]. — 2023. — URL: <https://econs.online/articles/details/syrevaya-bipolyarizatsiya-mira/> (дата обращения: 14.07.2026).
6. SIBUR Magazine. Стратегическое планирование закупок: как обеспечить бесперебойность поставок [Электронный ресурс]. — 2025. — URL: <https://magazine.sibur.ru/publication/workshop/strategicheskoe-planirovanie-zakupok-kak-obespechit-bespereboynost-postavok/> (дата обращения: 14.07.2026).
7. Investing.com. «Медвежи» риски для сырьевых рынков усиливаются: HSBC [Электронный ресурс]. — 2024. — URL: <https://ru.investing.com/news/commodities-news/article-2503752> (дата обращения: 14.07.2026).

Интеграция Process Mining и искусственного интеллекта для предиктивного управления бизнес-процессами

Пронин Кирилл Николаевич, руководитель группы разработки
ООО «Нейромед» (г. Москва)

Леонтьева Дарья Сергеевна, эксперт
Банк ВТБ (ПАО) (г. Москва)

В работе исследуются возможности интеграции технологий Process Mining и искусственного интеллекта для повышения эффективности анализа и управления бизнес-процессами организаций. Проведен обзор современных методов Process Mining, алгоритмов интеллектуального анализа данных и существующих подходов к предиктивному мониторингу процессов. Предложена концептуальная архитектура интеллектуальной системы, объединяющая обработку журналов событий, анализ процессов, модели машинного обучения и поддержку принятия управленческих решений. Показано, что применение интегрированного подхода позволяет прогнозировать задержки выполнения процессов, выявлять аномалии и риски нарушения SLA, а также повысить прозрачность и эффективность управления бизнес-процессами. Обоснованы преимущества использования event-driven архитектуры и потоковой обработки данных для построения масштабируемых корпоративных аналитических платформ. Перспективы дальнейших исследований связаны с разработкой объяснимых AI-моделей, применением генеративного искусственного интеллекта и созданием самоадаптирующихся BPM-систем.

Ключевые слова: Process Mining, искусственный интеллект, машинное обучение, бизнес-процессы, журналы событий, предиктивный мониторинг, BPM, event-driven архитектура, интеллектуальная аналитика, корпоративные информационные системы.

Введение

В настоящее время большинство организаций используют в рамках своей деятельности корпоративные информационные системы класса ERP, CRM, BPM, MES и другие цифровые решения, обеспечивающие автоматизацию и цифровизацию бизнес-процессов. Неотъемлемой частью функционирования указанных систем является формирование журналов событий (event logs) больших объемов данных, отражающих фактическое выполнение процессов, действия пользователей, временные характеристики операций и маршруты выполнения задач.

Несмотря на то, что с каждым годом общий уровень цифровизации повышается, управление бизнес-процессами во многих организациях по-прежнему строится преимущественно на регламентных моделях, экспертных оценках и ручном анализе данных. Использование такого подхода заметно замедляет получение актуальной информации о фактическом исполнении процессов, своевременное выявление отклонений, а также оперативное реагирование на возникающие риски и задержки.

Одной из современных технологий анализа бизнес-процессов, способной устранить имеющиеся недостатки устаревшего подхода, является технология Process Mining, представляющая собой совокупность методов построения, анализа и оптимизации процессов на основе журналов событий информационных систем [1]. В отличие от классических BPM-подходов, технология Process Mining позволяет исследовать реальные сценарии исполнения процессов, выявлять узкие места, нарушения регламентов и неэффективные схемы исполнения процессов

[2]. Различные методы и алгоритмы Process Mining подробно рассматриваются в работах Ван дер Аалста [1–3], являющегося одним из основоположников данного научного направления. В исследованиях [4–6] рассматриваются вопросы построения процессных моделей на основе журналов событий, а также методы проверки соответствий и автоматического обнаружения процессов. В работах [7, 8] исследуются вопросы применения интеллектуального анализа данных и методов машинного обучения для прогнозирования поведения бизнес-процессов.

Технология Process Mining активно развивается на зарубежном рынке, однако ее применение в отечественных организациях остается ограниченным. Главными факторами, создающими препятствие для распространения, являются фрагментированность корпоративных информационных систем, отсутствие унифицированных событийных моделей, недостаточный уровень зрелости процессов управления данными, а также сложность интеграции методов интеллектуального анализа в существующую ИТ-инфраструктуру организаций.

В последние годы также наблюдается активное развитие технологий искусственного интеллекта и машинного обучения, применение которых позволит существенно расширить возможности Process Mining. Интеграция методов искусственного интеллекта с системами анализа процессов обеспечит переход от ретроспективного анализа к предиктивному управлению процессами, включая прогнозирование временных задержек, выявление аномалий, обнаружение рисков нарушения SLA и интеллектуальную поддержку принятия управленческих решений [9, 10].

Целью настоящей работы является исследование возможностей интеграции методов Process Mining и искусственного интеллекта для повышения эффективности анализа и управления бизнес-процессами, а также разработка концептуального подхода к построению интеллектуальных систем предиктивного мониторинга процессов. В рамках исследования рассматриваются архитектурные подходы к интеграции Process Mining и AI-моделей, методы обработки журналов событий, а также сценарии применения алгоритмов машинного обучения для анализа и прогнозирования состояния бизнес-процессов.

Обзор существующих исследований

Технология Process Mining сформировалась как самостоятельное направление анализа бизнес-процессов на стыке таких технологий как Business Process Management (BPM), Data Mining и Process Analytics. Основополагающие принципы Process Mining были сформулированы Вилом Ван дер Аалстом (Wil van der Aalst) в работах [1–3], в которых предложены методы извлечения моделей процессов на основе журналов событий информационных систем.

Классический подход Process Mining включает три основных направления: process discovery, conformance checking и process enhancement [1]. Метод process discovery предназначен для автоматического построения модели процесса на основе журналов событий без предварительного описания процесса. В работах [4, 5] также предложены алгоритмы Alpha Miner, Heuristic Miner и Fuzzy Miner, предназначенные для восстановления структуры процессов по журналам событий. Однако указанные алгоритмы обладают рядом ограничений, связанных с чувствительностью к шумам данных, сложностью обработки неструктурированных процессов и ограниченной масштабируемостью при работе с высоконагруженными корпоративными системами. Метод conformance checking используется для сопоставления фактического выполнения процесса с его регламентной моделью, а process enhancement применяется для оптимизации и расширения существующих моделей процессов.

Существенное развитие технология Process Mining получила после появления специализированных программных платформ, таких как ProM, Celonis, Disco и PM4Py [6–8]. Упомянутые платформы позволяют на основании загруженных журналов событий визуализировать процессы, проанализировать временные характеристики выполнения операций, осуществить поиск bottleneck-участков или «узких мест» процесса и контроль соответствия процессов установленным регламентам. В исследованиях [9–11] рассматриваются также вопросы применения методов машинного обучения и искусственного интеллекта в задачах анализа бизнес-процессов. Использование AI-алгоритмов позволяет реализовать predictive process monitoring, включающий прогнозирование времени завершения процессов, вероятности воз-

никновения отклонений, оценку рисков нарушения SLA, а также автоматическое обнаружение аномалий в журналах событий.

Как было упомянуто ранее, одним из перспективных направлений совершенствования Process Mining является применение нейронных сетей и методов deep learning для анализа последовательностей событий процессов. В работах [12, 13] предлагается использование рекуррентных нейронных сетей (RNN) и архитектур класса LSTM для прогнозирования дальнейшего поведения процесса на основе исторических данных. Применение указанных методов позволяет повысить точность предиктивного анализа по сравнению с классическими статистическими моделями. В публикациях [14, 15] рассматриваются также подходы к интеграции Process Mining с интеллектуальными системами поддержки принятия решений. Авторы отмечают, что объединение методов анализа процессов и искусственного интеллекта позволяет перейти от ретроспективного анализа к адаптивному управлению бизнес-процессами в режиме реального времени.

Несмотря на значительное количество зарубежных исследований в области Process Mining, вопросы практической интеграции методов искусственного интеллекта с корпоративными BPM и ERP-системами остаются недостаточно исследованными. Кроме того, в существующих исследованиях недостаточно подробно рассматриваются вопросы построения единой архитектуры интеллектуального анализа процессов, объединяющей event-driven подходы, Process Mining и AI-модели в рамках единой корпоративной платформы управления процессами.

Таким образом, анализ существующих исследований показывает, что интеграция технологий Process Mining и искусственного интеллекта является перспективным направлением развития систем управления бизнес-процессами и требует дальнейшего исследования архитектурных, алгоритмических и прикладных аспектов построения интеллектуальных систем предиктивного анализа процессов.

Архитектура интеграции Process Mining и искусственного интеллекта

Современные корпоративные информационные системы формируют значительные объемы событийных данных, отражающих фактическое выполнение бизнес-процессов организации. Для реализации интеллектуального анализа процессов требуется построение интегрированной архитектуры, объединяющей технологии сбора событийных данных, методы Process Mining и алгоритмы искусственного интеллекта в рамках единой аналитической платформы. Пример обобщенной архитектуры системы интеллектуального анализа бизнес-процессов представлен на рисунке 1.

Основным элементом архитектуры является слой сбора и агрегации событийных данных или event collection layer. На данном уровне осуществляется получение жур-

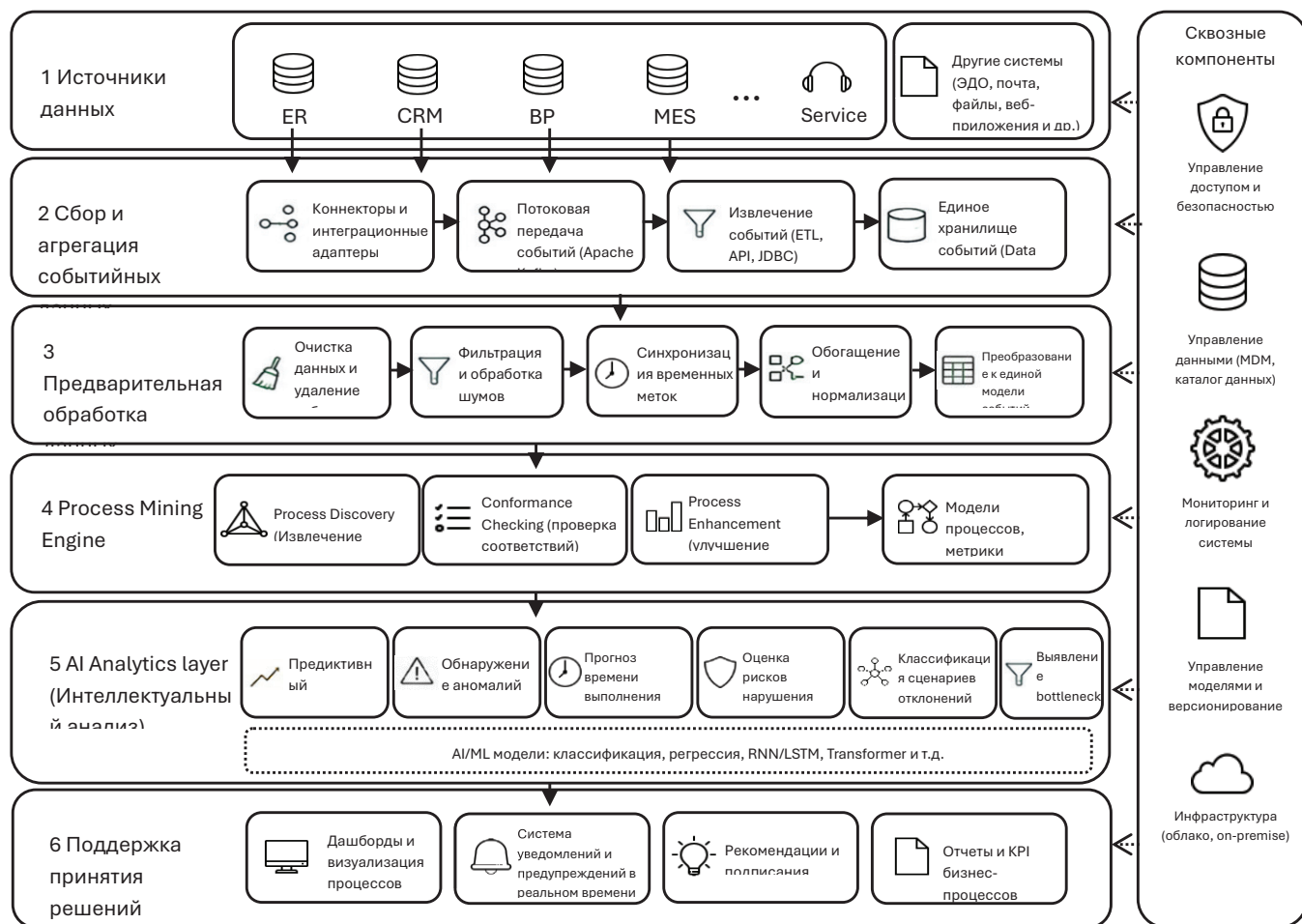


Рис. 1. Пример обобщенной архитектуры системы интеллектуального анализа бизнес-процессов

налов событий из корпоративных информационных систем организации. Каждое событие процесса должно содержать минимальный набор атрибутов: идентификатор процесса (Case ID), идентификатор операции (Activity ID), временную метку события (Timestamp), а также дополнительные параметры выполнения процесса [1].

Журнал событий бизнес-процесса может быть представлен в виде множества:

$$L = \{e_1, e_2, \dots, e_n\}, \quad (1)$$

где e_i — отдельное событие процесса, описываемое кортежем:

$$e_i = (c_i, a_i, t_i, r_i), \quad (2)$$

где

c_i — идентификатор экземпляра процесса (Case ID);

a_i — выполняемая операция (Activity);

t_i — временная метка события (Timestamp);

r_i — набор дополнительных атрибутов события (Resource, Cost, Status и др.).

Каждое событие характеризует факт выполнения операции в рамках определенного экземпляра процесса в конкретный момент времени и может быть дополнено контекстной информацией о ресурсе, стоимости, статусе и других параметрах. Последовательность событий для одного экземпляра процесса образует «таймлайн»:

$$\sigma_c = \langle e_{c,1}, e_{c,2}, \dots, e_{c,m_c} \rangle, \quad (3)$$

где m_c — количество событий в таймлайне экземпляра c , $e_{c,j} \in L$.

Модель процесса M , извлеченная из журнала событий L , может быть представлена в виде ориентированного графа (например, сети Петри):

$$M = (P, T, F, W, m_0), \quad (4)$$

где

$P = \{p_1, p_2, \dots, p_n\}$ — множество позиций;

$T = \{t_1, t_2, \dots, t_m\}$ — множество переходов (операций);

$F \subseteq (P \times T) \cup (T \times P)$ — множество дуг;

$W: F \rightarrow \mathbb{N}$ — функция весов дуг;

$m_0: P \rightarrow \mathbb{N}$ — начальная маркировка.

Качество соответствия (fitness) модели M журналу событий L может быть оценено метрикой:

$$fitness(M, L) = 1 - \frac{|L_{dev}|}{|L|}, \quad (5)$$

где L_{dev} — множество отклоняющихся (несоответствующих модели M) событий.

После получения журналов событий данные поступают в модуль предварительной обработки и нормализации, в рамках которого осуществляется фильтрация шумов, удаление дублирующих событий, синхронизация временных меток, а также преобразование данных к унифицированной модели процессов.

Следующим компонентом архитектуры является Process Mining Engine, обеспечивающий построение моделей процессов на основе журналов событий. В рамках данного модуля реализуются алгоритмы process discovery, conformance checking и process enhancement [2]. Результатом работы Process Mining Engine являются модели процессов, отражающие фактическую структуру выполнения бизнес-процессов организации. Полученные модели процессов и журналы событий передаются в модуль интеллектуального анализа (AI Analytics Layer), в котором применяются методы машинного обучения и искусственного интеллекта. Наиболее перспективными направлениями использования AI в задачах Process Mining являются:

- predictive process monitoring;
- anomaly detection;
- прогнозирование времени завершения процесса;
- выявление рисков нарушения SLA;
- интеллектуальная классификация сценариев выполнения процессов;
- автоматическое выявление bottleneck-участков.

Для решения указанных задач могут использоваться методы supervised learning, unsupervised learning, а также нейронные сети класса LSTM и Transformer [9–12].

Одним из ключевых элементов архитектуры является механизм предиктивного мониторинга процессов. На основе исторических данных полученных из журналов событий AI-модель формирует прогноз дальнейшего поведения процесса, включая вероятность отклонения от регламентного сценария, прогноз времени выполнения операции и вероятность возникновения ошибок. Результаты интеллектуального анализа передаются в подсистему визуализации и поддержки принятия решений (Decision Support Layer). На данном уровне реализуются интерактивные процессные дашборды, системы уведомлений, аналитические панели мониторинга и механизмы формирования рекомендаций для владельцев и участников процессов.

Интеграция методов искусственного интеллекта и Process Mining позволяет перейти от классического ретроспективного анализа процессов к интеллектуальному управлению бизнес-процессами в режиме реального времени. В отличие от традиционных BPM-систем, предлагаемый подход обеспечивает возможность динамического прогнозирования поведения процессов и автоматического выявления потенциальных отклонений еще до момента их фактического возникновения. Кроме того, использование событийной архитектуры обеспечивает возможность масштабирования системы интеллектуального анализа процессов и интеграции с современными платформами потоковой обработки данных, включая Apache Kafka, Apache Flink и Spark Streaming.

Таким образом, интеграция технологий Process Mining и искусственного интеллекта формирует основу построения интеллектуальных систем управления бизнес-процессами нового поколения, обеспечивающих адаптивное и предиктивное управление процессами организаций.

Методы обработки event logs и применение алгоритмов машинного обучения

Как было упомянуто ранее, основным источником данных для систем Process Mining являются журналы событий (event logs), формируемые корпоративными информационными системами в процессе выполнения бизнес-процессов. Такие журналы содержат информацию о последовательности выполнения операций, времени их выполнения, участниках процессов и других параметрах, характеризующих фактическое исполнение процессов. Как правило, журналы событий формируются внутри корпоративных информационных систем. Каждое событие процесса содержит минимальный набор атрибутов, включающий идентификатор экземпляра процесса, выполняемую операцию, временную метку события и сведения об исполнителе операции.

Журналы событий часто содержат неполные записи, дублирующиеся события, ошибки временных меток и несогласованные наименования операций. В связи с этим важным этапом интеллектуального анализа процессов является предварительная обработка событийных данных. К основным методам обработки журналов событий относятся удаление дублирующихся записей, фильтрация шумов, синхронизация временных меток, нормализация названий операций, а также объединение событий из различных информационных систем в единый журнал процессов. Дополнительно может выполняться фильтрация аномальных или неполных таймлайнов процессов, не пригодных для дальнейшего анализа. После предварительной обработки событийные данные преобразуются в формат, пригодный для применения алгоритмов машинного обучения. Для этого выполняется выделение признаков, характеризующих поведение процесса. В качестве таких признаков могут использоваться длительность выполнения операций, количество возвратов между этапами, число участников процесса, время ожидания между событиями, количество отклонений от регламентного сценария и другие параметры.

В интеллектуальном Process Mining применяются как классические методы машинного обучения, так и современные алгоритмы глубокого обучения. Для задач классификации процессов и выявления отклонений используются Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine и Logistic Regression. Для задач предиктивного мониторинга процессов наиболее эффективными являются модели анализа последовательностей событий. Такие модели позволяют прогнозировать дальнейшее поведение процесса на основе уже выполненных операций и накопленных исторических данных.

В последние годы широкое распространение получили рекуррентные нейронные сети и архитектуры класса Long Short-Term Memory (LSTM), обеспечивающие анализ последовательностей событий и выявление долгосрочных зависимостей между этапами процесса. Применение указанных моделей позволяет прогнозировать ве-

роятность нарушения SLA, оценивать ожидаемое время завершения процесса и выявлять потенциальные отклонения еще до момента их фактического возникновения. Перспективным направлением является также использование Transformer-архитектур и attention-механизмов для анализа сложных процессов с большим количеством вариантов исполнения. В отличие от классических рекуррентных сетей, Transformer-модели обеспечивают более эффективную обработку длинных последовательностей событий и позволяют учитывать взаимосвязи между удаленными этапами процесса.

Для автоматического обнаружения нетипичных сценариев выполнения процессов применяются методы unsupervised learning, включая кластеризацию, Isolation Forest и Autoencoder-модели. Использование данных методов позволяет выявлять аномалии в журналах событий, нарушения регламентов, подозрительные действия пользователей и нестандартные маршруты выполнения процессов.

В задачах прогнозирования времени завершения процессов используются методы регрессионного анализа и нейросетевые модели. Полученные прогнозы могут применяться для раннего обнаружения рисков нарушения SLA, динамического перераспределения ресурсов и автоматической корректировки маршрутов выполнения процессов.

Таким образом, применение методов машинного обучения существенно расширяет возможности Process Mining и позволяет перейти от анализа исторических данных к интеллектуальному прогнозированию поведения процессов и поддержке принятия управленческих решений в режиме реального времени.

Преимущества и ограничения интеграции Process Mining и искусственного интеллекта

Интеграция технологий Process Mining и искусственного интеллекта позволяет расширить возможности традиционного анализа бизнес-процессов за счет перехода от ретроспективного выявления проблем к предиктивному и адаптивному управлению процессами. В классических BPM-системах основное внимание уделяется описанию регламентных моделей процессов и контролю их исполнения. Однако фактическое выполнение процессов в корпоративных информационных системах часто отличается от формально описанных регламентов, что приводит к возникновению скрытых отклонений, задержек, повторных операций и неэффективных маршрутов выполнения задач. Использование Process Mining позволяет восстановить реальную модель процесса на основе журналов событий, сформированных в информационных системах организации. Это обеспечивает возможность объективного анализа фактического исполнения процессов без необходимости проведения длительных интервью, ручного моделирования и экспертной реконструкции бизнес-логики. В результате организация получает не предполагаемую, а фактическую картину выполнения про-

цесса, включая основные сценарии, исключения, узкие места и отклонения от нормативной модели.

Дополнение Process Mining методами искусственного интеллекта позволяет решать более широкий круг задач. В частности, AI-модели могут использоваться для прогнозирования времени завершения процесса, оценки вероятности нарушения SLA, выявления аномальных сценариев, классификации типов отклонений, а также формирования рекомендаций по дальнейшим действиям. Таким образом, система анализа процессов становится не только инструментом диагностики, но и средством поддержки принятия управленческих решений.

Одним из ключевых преимуществ предлагаемого подхода является повышение прозрачности бизнес-процессов. Руководители и владельцы процессов получают возможность анализировать не только итоговые показатели деятельности, но и внутреннюю структуру выполнения операций: последовательность действий, длительность этапов, частоту возвратов, количество ручных вмешательств и степень соответствия процесса установленным регламентам.

Вторым важным преимуществом является снижение времени обнаружения проблем. Традиционный анализ процессов, как правило, проводится постфактум и требует значительных трудозатрат со стороны аналитиков. В предлагаемой архитектуре отклонения могут выявляться автоматически на основе событийных данных, поступающих из корпоративных информационных систем. Это позволит сократить время между возникновением проблемы и принятием корректирующих мер.

Третьим преимуществом является возможность предиктивного управления процессами. В отличие от классического мониторинга, фиксирующего уже произошедшие отклонения, интеллектуальный анализ позволяет прогнозировать вероятность возникновения проблем до их фактического наступления. Например, система может заранее определить, что конкретный экземпляр процесса с высокой вероятностью превысит допустимое время выполнения или отклонится от стандартного маршрута.

Кроме того, интеграция Process Mining и AI способствует повышению качества управленческих решений. Рекомендации, формируемые системой, основываются не только на экспертном опыте, но и на статистическом анализе исторических данных, фактических сценариях выполнения процессов и выявленных закономерностях. Это особенно важно для сложных процессов, в которых количество возможных вариантов исполнения значительно превышает возможности ручного анализа. Практическая ценность предлагаемого подхода заключается также в возможности оптимизации операционных затрат. Выявление повторных операций, избыточных согласований, неэффективных маршрутов и bottleneck-участков позволяет определить направления автоматизации и реинжиниринга бизнес-процессов. В результате чего организация может сократить длительность выполнения процессов, уменьшить количество ручных операций и повысить общую производительность.

Вместе с тем интеграция Process Mining и искусственного интеллекта имеет ряд ограничений. Так одним из ограничений является зависимость качества анализа от полноты и корректности журналов событий. Если корпоративные информационные системы не фиксируют необходимые события либо содержат неполные, несогласованные или ошибочные данные, результаты Process Mining и AI-прогнозирования могут быть недостоверными. Также существенным ограничением является сложность унификации событийных данных, получаемых из различных информационных систем. В крупных организациях процессы часто проходят через несколько систем, каждая из которых имеет собственную структуру данных, формат хранения событий и правила идентификации экземпляров процесса. Это усложняет построение единого журнала событий и требует разработки дополнительного слоя нормализации данных. Помимо этого, необходимо учитывать и риск некорректной интерпретации результатов интеллектуального анализа. AI-модели способны выявлять статистические зависимости, однако такие зависимости не всегда отражают причинно-следственные связи. Поэтому результаты анализа должны рассматриваться как инструмент поддержки принятия решений, а не как полностью автономный механизм управления процессами. Отдельной проблемой стоит выделить необходимость обеспечения объяснимости AI-моделей. Для практического применения в корпоративной среде недостаточно только получить прогноз отклонения или нарушения SLA. Система должна предоставлять пользователю интерпретируемое объяснение причин прогноза, указывать факторы, повлиявшие на результат, и формировать понятные рекомендации для владельца процесса.

Таким образом, интеграция Process Mining и искусственного интеллекта позволяет существенно повысить прозрачность, управляемость и предсказуемость бизнес-процессов. Предлагаемый подход решает задачу перехода от статического описания процессов к их динамическому анализу и предиктивному управлению. При этом эффективность применения данного подхода напрямую зависит от качества событийных данных, зрелости ИТ-инфраструктуры организации и корректности выбранных методов машинного обучения.

Заключение

В данной работе рассмотрены вопросы интеграции технологий Process Mining и искусственного интеллекта для построения интеллектуальных систем анализа и предиктивного управления бизнес-процессами. Проведенный анализ существующих исследований показал, что современные методы Process Mining позволяют эффективно восстанавливать фактические модели процессов на основе журналов событий корпоративных информационных систем, выявлять отклонения от регламентов и анализировать структуру выполнения процессов.

Установлено, что интеграция методов искусственного интеллекта существенно расширяет возможности классического Process Mining. Использование алгоритмов машинного обучения позволяет перейти от ретроспективного анализа процессов к предиктивному мониторингу и интеллектуальной поддержке принятия решений. В частности, применение AI-моделей обеспечивает возможность прогнозирования времени выполнения процессов, выявления аномалий, оценки рисков нарушения SLA и автоматического обнаружения bottleneck-участков.

В работе предложена архитектура интеграции Process Mining и искусственного интеллекта, включающая слой сбора событийных данных, предварительной обработки event logs, анализа процессов, интеллектуальной аналитики и поддержки принятия решений. Предлагаемый подход обеспечивает возможность построения единой аналитической платформы для мониторинга и управления бизнес-процессами в режиме реального времени.

Показано, что применение event-driven архитектуры и потоковой обработки событий позволяет обеспечить масштабируемость системы и интеграцию с современными корпоративными информационными платформами. Использование технологий Apache Kafka, потоковой аналитики и AI-моделей формирует основу для создания интеллектуальных систем управления процессами нового поколения.

Вместе с тем установлено, что эффективность применения интеллектуального Process Mining напрямую зависит от качества журналов событий, уровня зрелости корпоративной ИТ-инфраструктуры и степени унификации событийных данных. Существенными ограничениями являются фрагментированность информационных систем, неполнота event logs и сложность интерпретации результатов работы AI-моделей.

Практическая значимость предлагаемого подхода заключается в повышении прозрачности бизнес-процессов, сокращении времени обнаружения отклонений, снижении операционных затрат и повышении эффективности управленческих решений. Наиболее перспективными направлениями применения интеллектуального Process Mining являются системы управления заявками, процессы технической поддержки, медицинские информационные системы, производственные процессы и корпоративные системы электронного документооборота.

Действительно, интеграция технологий Process Mining и искусственного интеллекта можно назвать перспективным направлением развития интеллектуальных систем управления бизнес-процессами и основой для перехода от статического BPM-подхода к адаптивному и предиктивному управлению процессами организации. Перспективы дальнейших исследований связаны с разработкой explainable AI-моделей для Process Mining, применением генеративного искусственного интеллекта в задачах автоматической оптимизации процессов, а также построением self-adaptive BPM-систем, способных автоматически изменять сценарии выполнения процессов в зависимости от текущего состояния бизнес-среды.

Литература:

1. van der Aalst W. M. P. Process Mining: Data Science in Action. 2nd ed. Berlin: Springer, 2016. 467 p.
2. van der Aalst W. M. P. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Berlin: Springer, 2011. 352 p.
3. van der Aalst W. M. P., van Hee K. M. Workflow Management: Models, Methods, and Systems. Cambridge: MIT Press, 2002. 386 p.
4. van der Aalst W. M. P., Weijters A. J. M. M., Maruster L. Workflow Mining: Discovering Process Models from Event Logs // IEEE Transactions on Knowledge and Data Engineering. 2004. Vol. 16, no. 9. P. 1128–1142.
5. Leemans S. J. J., Fahland D., van der Aalst W. M. P. Discovering Block-Structured Process Models from Event Logs — A Constructive Approach // Application and Theory of Petri Nets and Concurrency. Berlin: Springer, 2013. P. 311–329.
6. Gv/bnther C. W., van der Aalst W. M. P. Fuzzy Mining — Adaptive Process Simplification Based on Multi-perspective Metrics // Business Process Management. BPM 2007. Lecture Notes in Computer Science, vol. 4714. Berlin: Springer, 2007. P. 328–343.
7. Berti A., van Zelst S. J., van der Aalst W. M. P. Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science // arXiv preprint arXiv:1905.06169. 2019.
8. Reinkemeyer L. Process Mining in Action: Principles, Use Cases and Outlook. Cham: Springer, 2020. 279 p.
9. Tax N., Verenich I., La Rosa M., Dumas M. Predictive Business Process Monitoring with LSTM Neural Networks // Advanced Information Systems Engineering. CAiSE 2017. Lecture Notes in Computer Science, vol. 10253. Cham: Springer, 2017. P. 477–492.
10. Evermann J., Rehse J.-R., Fettke P. Predicting Process Behaviour Using Deep Learning // Decision Support Systems. 2017. Vol. 100. P. 129–140.
11. van der Aalst W. M. P. Data Science in Action // Process Mining Handbook. Cham: Springer, 2022. P. 3–23.
12. Adams J. N., van der Aalst W. M. P. Precision and Fitness in Object-Centric Process Mining // arXiv preprint arXiv:2110.05375. 2021.
13. van der Aalst W. M. P., Brockhoff T., Ghahfarokhi A. F., Pourbafrani M., Uysal M. S., van Zelst S. Removing Operational Friction Using Process Mining: Challenges Provided by the Internet of Production // arXiv preprint arXiv:2107.13066. 2021.
14. IEEE Task Force on Process Mining. Process Mining Manifesto // Business Process Management Workshops. Berlin: Springer, 2012. P. 169–194.
15. Dumas M., Rosa M. La, Mendling J., Reijers H. A. Fundamentals of Business Process Management. 2nd ed. Berlin: Springer, 2018. 527 p.

Особенности использования нейронных сетей в управлении беспилотным автомобилем

Селезнёв Александр Игоревич, ассистент;

Селезнёв Игорь Львович, кандидат технических наук, доцент

Белорусский государственный университет информатики и радиоэлектроники (г. Минск, Беларусь)

В статье рассматривается использование нейронных сетей в управлении беспилотным автомобилем. Анализируются преимущества интеграции нейронных сетей при решении задач в эксплуатации беспилотных автомобилей, обсуждаются архитектурные особенности применения нейронных сетей.

Ключевые слова: нейронные сети, беспилотный автомобиль.

В настоящее время нейронные сети (НС) играют ключевую роль в развитии технологий беспилотного транспорта, обеспечивая интеллектуальное и адаптивное управление в сложных, динамически изменяемых условиях внешней среды. Применение НС охватывает широкий спектр задач — от восприятия окружающей среды до принятия высокоуровневых решений по управлению движением. Способность нейронных сетей к обучению на больших массивах данных, адаптации к динамическим условиям

и принятию решений в реальном времени делает их предпочтительными для применения в беспилотном транспорте.

Внедрение нейронных сетей в системы управления беспилотным автомобилем (БА) обусловлено их возможностями для решения комплекса фундаментальных задач, с которыми сталкиваются автономные транспортные средства в реальных условиях эксплуатации:

1. Обработка мультимодальных сенсорных данных. Беспилотный автомобиль получает информацию от раз-

нородных сенсоров, каждый из которых генерирует специфические данные:

- камеры — двумерные изображения высокой детализации (RGB, инфракрасные);
- лидары — трёхмерные облака точек с координатами (x,y,z) и интенсивностью отражения;
- радары — данные о скорости объектов и дистанции;
- ультразвуковые датчики — ближняя зона обнаружения (до 5 м);
- GPS (Global Positioning System)/IMU (Inertial Measurement Unit) — геопозиционные данные и параметры движения (ускорение, угловая скорость).

Применение НС позволяет интегрировать разнородные данные в единое семантическое представление [1], компенсировать ограничения отдельных сенсоров (например, лидар «не видит» цвет, камера теряет детализацию в тумане) [2], выявлять скрытые корреляции между различными модальностями (например, сопоставление формы объекта по информации от камеры и его скорости от радара) [3].

2. Работа в условиях неопределённости и стохастичности среды. Дорожная среда характеризуется непредсказуемостью по следующим факторам:

- неопределённое поведение участников движения. Пешеходы могут внезапно выйти на проезжую часть, водители — резко сменить полосу;
- динамические препятствия (например, животные, падающие предметы, временные конструкции);
- изменяющиеся внешние условия среды. К ним относятся погодные явления (например, дождь, снег, туман), смена освещения (например, закат, туннели), сезонные изменения (например, опавшая листва, сугробы).

Применение нейронных сетей позволяет обобщать различные ситуации за счет обучения, производить вероятностное прогнозирование (например, оценка вероятности перехода пешехода через дорогу) и адаптироваться к новым сценариям.

3. Требования к времени реакции. Критические ситуации на дороге требуют немедленного реагирования:

- обнаружение пешехода на скорости 60 км/ч с учетом времени на торможение не более чем 150 мс;
- уклонение от внезапно появившегося препятствия — время реагирования должно быть не более чем 200 мс.

Применение НС позволяет производить своевременное реагирование за счет использования существующих оптимизаций под аппаратные ускорители (GPU — Graphics Processing Unit, TPU — Tensor Processing Unit, FPGA — Field-Programmable Gate Array) для обработки данных в реальном масштабе времени, использования легких архитектур сверточных нейросетей (MobileNet, EfficientNet) для низкократных вычислений [4] и конвейерной обработки с параллельным выполнением этапов (обнаружение, классификация, прогнозирование) [5].

4. Семантическое понимание сцены. Для безопасного движения необходимо не просто «видеть» объекты, но и интерпретировать контекст:

- различать типы перекрёстков (регулируемые/нерегулируемые);
- распознавать намерения водителей (например, мигание фонаря поворота, замедление);
- учитывать социальные нормы (например, пропуск пешехода на переходе типа «зебра»).

НС позволяют производить семантическую сегментацию (разделение изображения на классы — дорога, обочина, разметка) и instance-сегментацию (идентификация отдельных экземпляров объектов); использование нейронных сетей на основе графов (Graph Neural Network, GNN) позволяет моделировать взаимодействия между участниками движения [6].

5. Планирование и принятие решений. Беспилотный автомобиль должен не только воспринимать среду, но и вырабатывать стратегию движения:

- выбирать оптимальную траекторию движения с учётом препятствий;
- прогнозировать последствия при совершении манёвров (например, риск столкновения при перестроении движения с одной полосы на другую);
- соблюдать баланс между безопасностью и комфортом (например, плавное торможение и экстренное уклонение).

Рекуррентные нейронные сети (LSTM — Long Short-Term Memory, GRU — Gated Recurrent Units) позволяют моделировать временные зависимости [7], а обучение с подкреплением (reinforcement learning) позволяет оптимизировать политики управления. Также возможно создание гибридных систем, сочетающих НС с классическими алгоритмами (алгоритм поиска по первому наилучшему совпадению на графе A*, RRT — Rapidly-exploring Random Tree) [8].

6. Устойчивость к атакам и аномалиям на сенсоры БА:

- состязательные атаки (adversarial attacks, например, наклейки на дорожные знаки, имитирующие стоп-сигнал);
- аппаратные сбои (например, зашумленное изображение от камеры, потеря сигнала GPS);
- неопределённые объекты (аномальные объекты, не встречавшиеся в обучающих данных).

В этом случае роль НС состоит в разработке робастных (устойчивых) архитектур (например, с механизмами внимания) [9], ансамблевых методов для повышения надёжности [10], мониторинга уверенности модели (отказ от действий при низкой уверенности) [11].

7. Масштабируемость и адаптивность. Беспилотные системы должны работать в разных географических и инфраструктурных условиях:

- городские улицы и загородные шоссе;
- разные стандарты разметки и знаков (например, в Европе, Азии);
- изменение дорожной инфраструктуры (например, ремонт, новые развязки).

В таких случаях использование нейронных сетей обуславливается их исключительной адаптивностью за счет

различных видов обучения: трансферного (адаптация моделей к новым регионам), онлайн обучения (корректировка весов модели на основе новых данных) [12], мета-обучения (быстрое приспособление к неизвестным сценариям) [13].

Архитектурная роль нейронных сетей в системе управления беспилотным автомобилем заключается в том, что НС выступают перцепционным ядром системы, выполняя многоуровневую обработку сенсорных данных, структура которой приведена на рисунке 1.

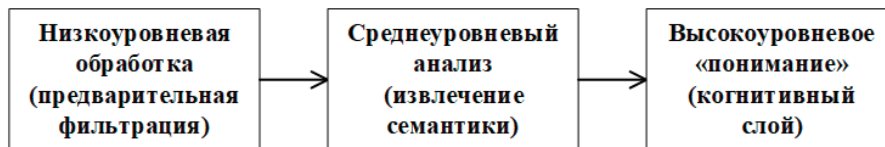


Рис. 1. Основные структурные уровни обработки сенсорных данных

Ключевой задачей нейронных сетей является преобразование «сырых» сигналов сенсоров в семантическое представление окружения, пригодное для принятия решений. На рисунке 1 приведены следующие основные структурные уровни обработки сенсорных данных нейронными сетями:

1. Низкоуровневая обработка (предварительная фильтрация). На этом уровне НС решают задачи первичной очистки и нормализации данных:

- фильтрация шумов. Свёрточные нейронные сети (Convolutional Neural Network, CNN) подавляют случайные артефакты в данных лидаров, камер и радаров;
- калибровка сенсоров. НС выравнивают данные разнородных датчиков (камеры, лидары, IMU) в единую координатную систему, компенсируя геометрические искажения оптики, временные задержки между сенсорами, температурные дрейфы;
- нормализация сигналов. Приведение данных к единому масштабу (например, нормализация яркости изображений или амплитуды радарных отражений).

Для низкоуровневой обработки используются следующие архитектуры: U-Net, автоэнкодеры, CNN с малыми ядрами (3×3).

2. Среднеуровневый анализ (извлечение семантики). Здесь НС выявляют структурные элементы окружения:

- обнаружение объектов. Сети типа YOLO, SSD или Faster R CNN локализуют транспортные средства, пешеходов, дорожные знаки и разметку [14]. Результатом их работы является набор ограничивающих рамок (bounding boxes) с вероятностями классов;
- сегментация сцен. Полносвёрточные сети (Fully Convolutional Network, FCN) или Transformer-архитектуры (SegFormer) разделяют изображение на пиксели с метками: дорога, обочина, здания, небо [15]. Результатом их работы является маска сегментации;
- оценка глубины. Монокулярные сети (например, Monodepth2) восстанавливают карту глубин из одиночного изображения [16].

Ключевыми технологиями среднеуровневого анализа являются CNN с пирамидальной обработкой (Feature Pyramid Network, FPN), attention-механизмы (механизмы внимания), мультимодальная интеграция (камера + лидар).

3. Высокоуровневое «понимание» (когнитивный слой). На этом уровне НС интерпретируют динамику изменения окружения и планируют действия:

- прогнозирование поведения. Рекуррентные сети (LSTM, GRU) или нейронные сети на основе графов моделируют траектории объектов [17,18];
- планирование манёвров. НС оценивают варианты действий (ускорение, поворот, торможение) с учётом правил дорожного движения, безопасности (дистанция, время до столкновения), комфорта пассажиров. Для этого используются сети ценностных функций (Q networks) и имитационные модели поведения («end to end» подходы, например ChauffeurNet) [19];
- понимание контекста. Сети-трансформеры «анализируют» долгосрочные зависимости: приоритеты на перекрёстках, намерения пешеходов (например, готовность перейти дорогу), аномалии (аварии, ремонтные работы).

Архитектурами для высокоуровневого «понимания» являются гибридные системы (классические алгоритмы в связке с НС), модели мира (world models), обучение с подкреплением (reinforcement learning).

Для интеграции в общую систему управления БА нейронные сети взаимодействуют с другими модулями посредством:

- sensor fusion (слияние данных с датчиков). Объединение данных камер, лидаров и радаров на уровне признаков (early fusion) или решений (late fusion);
- локализацию и SLAM (Simultaneous Localization and Mapping) для коррекции позиции автомобиля по семантическим ориентирам (знаки, разметка);
- модуль управления, который обеспечивает передачу высокоуровневых команд (например, «сменить полосу») в контроллер движения.

Без использования НС крайне сложно достичь требуемой гибкости (адаптация к новым городам/дорогам) и надёжности (работа в условиях частичной потери данных) в эксплуатации беспилотного автомобиля.

Развитие нейронных сетей для БА тесно связано с общей эволюцией искусственного интеллекта:

- в 2010-е годы был совершен прорыв в свёрточных сетях, который позволил достичь высокой точности в распознавании объектов [20];

— в 2020-е годы появились сети-трансформеры и фундаментальные модели (foundation models), что открыло возможности к комплексному «пониманию» сцены дорожного движения [21];

— в настоящее время происходит интеграция нейросетей с физикой движения и логическими системами для обеспечения формальной верификации безопасности [22].

Несмотря на прогресс, внедрение нейронных сетей в беспилотные системы сталкивается с рядом серьезных проблем:

— проблема «чёрного ящика». Отсутствие прозрачности в принятии решений затрудняет юридическую ответственность [23];

— катастрофическая забывчивость (catastrophic forgetting). Переобучение на новых данных может разрушить ранее приобретённые навыки [24];

— уязвимость к состязательным атакам. Незначительные изменения в входных данных (например, наклейки на знаках) способны вызвать критические ошибки [25].

Выводы

Нейронные сети выступают не просто инструментом, а системной основой для перехода к полностью автономному транспорту. Их роль в управлении беспилотным автомобилем выходит за рамки отдельных задач распознавания или планирования — они формируют единую систему, где данные от сенсоров трансформируются в безопасные и эффективные управляющие воздействия, что делает возможным создание автономных автомобилей для выполнения широкого спектра задач с надлежащим уровнем безопасности.

Литература:

1. Chen, X. Multi View 3D Object Detection Network for Autonomous Driving / X. Chen, H. Ma, J. Wan, B. Li, T. Xia // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — P. 1907–1915.
2. Janai, J. Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art / J. Janai, F. Güney, A. Behlmann, V. Ranjan, A. Geiger // Foundations and Trends in Computer Graphics and Vision. — 2020. — Vol. 12, No. 1. — P. 1–153.
3. Simegnew, A. A Comprehensive Survey of Deep Learning Multisensor Fusion based 3D Object Detection for Autonomous Vehicles / A. Simegnew // Electronics. — 2022. — Vol. 11, No. 16. — Art. 2567. — 37 p.
4. Howard, A. G. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications / A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam // arXiv:1704.04861. — 2017.
5. Goodfellow, I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. — Cambridge, MA: MIT Press, 2016. — 800 p.
6. Fisac, J. F. A General Safety Framework for Learning Based Control in Robotics and Autonomous Systems / J. F. Fisac, N. Fridovich Keil, J. Harrison, A. Bajcsy, D. S. Brown, A. Dragan, A. Sastry, C. Tomlin, S. Sastry // IEEE Transactions on Automatic Control. — 2022. — Vol. 67, No. 5. — P. 2235–2250.
7. Samek, W. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning / W. Samek, T. Wiegand, G. Montavon. — Berlin: Springer, 2019. — 320 p.
8. Al-Ansarry S., Al-Darraj S. Hybrid RRT-A*: An Improved Path Planning Method for an Autonomous Mobile Robot // International Journal of Electrical and Electronic Engineering. — 2021. — Vol. 17, № 1. — P. 13.
9. Athalye, A. Synthesizing Robust Adversarial Examples / A. Athalye, N. Carlini, D. Wagner // Proceedings of the 35th International Conference on Machine Learning (ICML). — 2018. — Vol. 80. — P. 284–293.
10. Mallya, A. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning / A. Mallya, S. Lazebnik // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2018. — P. 1860–1870.
11. Building confidence in autonomous car safety // Novatel: [сайт]. — URL: <https://blog.novatel.com/building-confidence-in-autonomous-car-safety/> (дата обращения: 28.06.2026)
12. Shalev Shwartz, S. Safe, Multi Agent, Reinforcement Learning for Autonomous Driving / S. Shalev Shwartz, S. Shammah, A. Shashua // arXiv:1610.03295. — 2016.
13. Koopman, P. Challenges in Autonomous Vehicle Testing and Validation / P. Koopman, M. Wagner // SAE International Journal of Transportation Safety. — 2016. — Vol. 4, No. 1. — P. 7–14.
14. Dakari D., Daniel C. A Comparative Analysis of Modern Object Detection Algorithms: YOLO vs. SSD vs. Faster RCNN // Information Technology Engineering Journal (ITEJ). — 2023. — Vol. 8, № 2. — P. 96–106.
15. Wang P., Chen P., Yuan Y., Liu D., Huang Z., Hou X., Cottrell G. Understanding Convolution for Semantic Segmentation // Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV). — 2018. — P. 1451–1460.
16. Godard C., Mac Aodha O., Brostow G. J. Unsupervised Monocular Depth Estimation with Left Right Consistency // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — P. 6602–6611.
17. Cai Y., Sarkar A., Zain J. M., Bhar A., Noorwali A., Othman K. M. Leveraging LSTM and GRU-Based Deep Neural Coordination in Intelligent Transportation to Strengthen Security in the Internet of Vehicles // International Journal of Machine Learning and Cybernetics. — 2025. — Vol. 16. — P. 2431–2467.

18. Rahmani S., Baghbani A., Bouguila N., Patterson Z. Graph Neural Networks for Intelligent Transportation Systems: A Survey // IEEE Transactions on Intelligent Transportation Systems. — 2023. — Vol. 24, no. 10. — P. 9276–9298.
19. Cai P., Wang H., Sun Y., Liu M. DQ GAT: Towards Safe and Efficient Autonomous Driving With Deep Q Learning and Graph Attention Networks // IEEE Transactions on Intelligent Transportation Systems. — 2022. — Vol. 23, No. 11. — P. 20134–20145.
20. Krizhevsky, A. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G. E. Hinton // Advances in Neural Information Processing Systems (NIPS). — 2012. — Vol. 25. — P. 1097–1105.
21. Dosovitskiy, A. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale / A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby // International Conference on Learning Representations (ICLR). — 2021.
22. Fisac, J. F. A General Safety Framework for Learning Based Control in Robotics and Autonomous Systems / J. F. Fisac, N. Fridovich Keil, J. Harrison, A. Bajcsy, D. S. Brown, A. Dragan, A. Sastry, C. Tomlin, S. Sastry // IEEE Transactions on Automatic Control. — 2022. — Vol. 67, No. 5. — P. 2235–2250.
23. Samek, W. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning / W. Samek, T. Wiegand, G. Montavon. — Berlin: Springer, 2019. — 320 p.
24. Kirkpatrick, J. Overcoming Catastrophic Forgetting in Neural Networks / J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell // Proceedings of the National Academy of Sciences (PNAS). — 2017. — Vol. 114, No. 13. — P. 3420–3425.
25. Athalye, A. Synthesizing Robust Adversarial Examples / A. Athalye, N. Carlini, D. Wagner // Proceedings of the 35th International Conference on Machine Learning (ICML). — 2018. — Vol. 80. — P. 284–293.

ТЕХНИЧЕСКИЕ НАУКИ

Системы фильтрации топлива в дизельных двигателях сельскохозяйственной техники

Шутов Вячеслав Максимович, студент
Пензенский государственный аграрный университет

В статье проведён обзор схем фильтрации топлива в дизельных двигателях сельскохозяйственной техники.

Ключевые слова: топливная система дизеля, фильтрация топлива, сельскохозяйственная техника.

Эксплуатация дизельного транспорта в аграрной сфере характеризуется рядом специфических условий, существенно отличающихся от типовых режимов работы автомобильной техники в городской или магистральной среде. К числу определяющих факторов относятся повышенная запылённость в полевых условиях, значительные сезонные колебания температур, удалённость от стационарных сервисных центров, а также высокая интенсивность использования техники в периоды посевных и уборочных кампаний. В совокупности эти обстоятельства предъявляют повышенные требования к надёжности и устойчивости ключевых систем силовой установки, в первую очередь — к топливной аппаратуре дизельного двигателя.

В этих условиях система фильтрации топлива перестаёт быть вспомогательным элементом и становится одним из ключевых факторов обеспечения долговечности и безотказности работы дизельного двигателя в сельскохозяйственном производстве. Эффективность фильтрации напрямую определяет степень защиты прецизионных пар от воздействия загрязнителей, а выбор оптимальной схемы очистки и соблюдение регламентов обслуживания позволяют минимизировать риски преждевременного выхода из строя дорогостоящих компонентов.

В современных дизельных двигателях сельскохозяйственной техники реализуется многоступенчатый подход к очистке топлива, обусловленный необходимостью последовательного удаления загрязнителей различного характера и размера. Традиционная схема фильтрации, применяемая на большинстве тракторов и комбайнов, предполагает двухступенчатую очистку: на первом этапе осуществляется грубая очистка топлива от крупных механических примесей и частичного отделения свободной воды, на втором — тонкая очистка, обеспечивающая задержание мелкодисперсных частиц и остаточной эмульсионной воды. Такая структура обусловлена различием

в функциональных задачах ступеней: фильтр грубой очистки (часто совмещённый с отстойником) снижает нагрузку на последующие элементы системы, а фильтр тонкой очистки обеспечивает соответствие качества топлива требованиям прецизионной топливной аппаратуры [3].

Фильтр грубой очистки, как правило, представляет собой сетчатый или щелевой элемент либо отстойную камеру, где отделение крупных частиц и свободной воды происходит за счёт гравитационного осаждения. В ряде конструкций предусматривается ручной слив отстоя, что позволяет оперативно удалять накопившуюся воду и тяжёлые фракции. Эффективность данной ступени оценивается прежде всего по способности задерживать частицы размером свыше 50–100 мкм и отделять значительную долю свободной воды, снижая тем самым риск мгновенного засорения фильтров тонкой очистки.

На ступени тонкой очистки применяются фильтрующие элементы из бумаги, синтетических волокон либо комбинированные конструкции с коагулирующими слоями. Их задача — удерживать частицы микронного диапазона (обычно 3–10 мкм), способные вызывать абразивный износ прецизионных пар, а также способствовать отделению мелкодисперсной эмульсионной воды за счёт коагуляции капель. В современных решениях всё чаще используются гидрофобные материалы и мембранные покрытия, препятствующие прохождению воды при сохранении пропускной способности по топливу [1].

Дальнейшее развитие схем фильтрации связано с переходом к каскадным (многоступенчатым) системам, в которых последовательно устанавливаются несколько ступеней с дифференцированными характеристиками. Согласно данным исследований, применение трёхступенчатой каскадной схемы позволяет повысить степень очистки топлива до 99,9 % за счёт поэтапного удаления загрязнителей различной природы: на первой ступени —

крупные частицы и свободная вода, на второй — мелко-дисперсные примеси и коагулированные капли воды, на третьей — остаточные микрочастицы и тонкодисперсная эмульсия [3]. Такая компоновка особенно актуальна для сельскохозяйственной техники, эксплуатируемой в условиях повышенной запылённости и влажности, поскольку обеспечивает более равномерное распределение нагрузки между ступенями и увеличивает общий ресурс системы.

Конструктивно каскадные схемы могут включать дополнительные функциональные элементы — например, фильтры-влагоотделители с полимерными адсорбентами или гидродинамические фильтры с сетчатыми гидрофобными перегородками. Последние позволяют не только механически задерживать примеси, но и управлять поведением водных включений за счёт гидродинамических эффектов, повышая эффективность отделения эмульсионной воды даже при изменяющихся режимах прокачки топлива [2, 6]. Важным преимуществом таких решений является возможность непрерывной регенерации или упрощённого обслуживания в полевых условиях, что соответствует требованиям эксплуатации сельхозтехники.

Анализ схем фильтрации топлива показывает закономерную эволюцию решений — от простых двухступенчатых систем к сложным каскадным конфигурациям, ориентированным на комплексную очистку топлива в тяжёлых условиях эксплуатации сельхозтехники. Классическая двухступенчатая схема (грубая очистка — тонкая очистка) остаётся востребованной благодаря своей простоте и ремонтпригодности, однако её ресурс и эффективность существенно ограничены при работе в условиях повышенной запылённости и высокой обводнённости топлива. В таких режимах нагрузка на фильтр тонкой

очистки резко возрастает, а риск прорыва загрязнений в топливную аппаратуру увеличивается.

Переход к каскадным (многоступенчатым) схемам позволяет нивелировать эти недостатки за счёт дифференцированного удаления загрязнителей: на ранних ступенях задерживаются крупные частицы и свободная вода, на последующих — мелкодисперсные примеси и эмульсионная вода. Как показывают исследования, применение трёхступенчатой каскадной схемы обеспечивает степень очистки до 99,9 %, что существенно повышает защиту прецизионных узлов топливной аппаратуры [3].

Важным фактором эффективности каскадных решений становится не только количество ступеней, но и грамотный подбор функциональных элементов для каждой из них. Использование гидрофобных материалов, мембранных покрытий, коагулирующих слоёв и гидродинамических фильтров позволяет целенаправленно управлять процессами отделения воды и улавливания мелко-дисперсных частиц [1, 2, 6]. При этом критически важна правильная последовательность установки элементов: нарушение компоновки (например, размещение фильтра тонкой очистки до водоотделителя) способно свести на нет преимущества многоступенчатой схемы.

Таким образом, современные схемы фильтрации должны проектироваться как сбалансированные системы, где каждая ступень решает конкретную задачу, а общая компоновка учитывает гидравлические характеристики топливной магистрали и реальные эксплуатационные режимы сельхозтехники. Это позволяет не только повысить степень очистки топлива, но и увеличить общий ресурс фильтрующей системы, снизив тем самым эксплуатационные затраты и риски отказов топливной аппаратуры.

Литература:

1. Новопашин Л. А., Лескин Р. Е. Результаты исследований фильтрующего элемента с мембранным напылением // НТВТСвАПК. 2019. № 3 (3). URL: <https://cyberleninka.ru/article/n/rezultaty-issledovaniy-filtruyuschego-elementa-s-membrannym-napyleniem> (дата обращения: 13.07.2026).
2. Улюкина Е. А., Орешников А. В., Шарыкин Ф. Е. Экспериментальное исследование эффективности гидродинамического фильтрования дизельного топлива // Агроинженерия. 2023. № 2. URL: <https://cyberleninka.ru/article/n/eksperimentalnoe-issledovanie-effektivnosti-gidrodinamicheskogo-filtrovaniya-dizelnogo-topliva> (дата обращения: 13.07.2026).
3. Коваленко В. П., Улюкина Е. А., Липаева М. А. Обеспечение чистоты топлив и масел при эксплуатации сельскохозяйственной техники // Агроинженерия. 2015. № 3 (67). URL: <https://cyberleninka.ru/article/n/obespechenie-chistoty-topliv-i-masel-pri-ekspluatatsii-selskohozyaystvennoy-tehniki> (дата обращения: 13.07.2026).
4. Новичков, А. В. Улучшение очистки топлива в топливной системе сельскохозяйственных тракторов использованием фильтра-влагоотделителя: дис. ... канд. техн. наук: 05.20.03 / А. В. Новичков.
5. Овчинников, О. С. Фильтр-водоотделитель для дизельных двигателей автомобилей / О. С. Овчинников. — Текст : непосредственный // Молодой ученый. — 2009. — № 12 (12). — С. 75–77. — URL: <https://moluch.ru/archive/12/949> (дата обращения: 13.07.2026).

МЕДИЦИНА

Оценка организации и качества амбулаторной медицинской помощи в первичном звене здравоохранения: взгляд медицинских работников

Ильясов Артикбай Серикович, студент магистратуры
Казахстанский медицинский университет «ВШОЗ» (г. Алматы, Казахстан)

Серик Майра Сериккызы, ассистент;
Иманбек Багымжан Кенеснызы, ассистент
Казахский национальный медицинский университет имени С. Д. Асфендиярова (г. Алматы, Казахстан)

Научный руководитель: Маханбеткулова Динара Нургалиевна, PhD, ассоциированный профессор
Казахстанский медицинский университет «ВШОЗ» (г. Алматы, Казахстан)

Первичная медико-санитарная помощь (ПМСП) является основой системы здравоохранения и играет ключевую роль в обеспечении доступности и качества медицинской помощи населению. В условиях реформирования системы здравоохранения и внедрения обязательного социального медицинского страхования (ОСМС) в Республике Казахстан особую актуальность приобретает изучение вопросов повышения эффективности организации амбулаторной медицинской помощи в первичном звене здравоохранения, а также выявление факторов, влияющих на качество и доступность медицинских услуг населению. Цель исследования — оценить организацию и качество амбулаторной медицинской помощи в ПМСП на основе опроса медицинских работников коммунального государственного предприятия на праве хозяйственного ведения (КГП на ПХВ) «Больница города Приозерск» Управления здравоохранения Карагандинской области (УЗКО). В результате исследования выявлены ключевые проблемы функционирования ПМСП: недостаточное финансирование, кадровый дефицит и высокая рабочая нагрузка, для решения которых предложены такие приоритетные меры, как повышение заработной платы, увеличение кадрового состава и повышение квалификации персонала.

Ключевые слова: ОСМС, профилактика, первичная медико-санитарная помощь, Казахстан, оптимизация.

Введение

Первичная медико-санитарная помощь является фундаментом системы здравоохранения и определяет доступность, своевременность и качество медицинской помощи населению [1]. В условиях глобальных вызовов — роста хронических заболеваний, демографического старения и дефицита медицинских кадров — вопросы оценки эффективности ПМСП приобретают особую актуальность [2].

В Казахстане реформирование системы здравоохранения последовательно направлено на усиление первичного звена. Тем не менее, несмотря на проводимые преобразования, в ПМСП сохраняются системные проблемы: кадровый дефицит, высокая рабочая нагрузка, недостаточное ресурсное обеспечение и ограниченное внедрение цифровых технологий [3]. Мнение самих медицинских работников является ценным источником информации для объективной оценки состояния первичного звена, однако остается недостаточно изученным на уровне отдельных медицинских организаций. Изучение мнения сотрудников ПМСП позволяет выявить организационные и кадровые проблемы, влияющие на эффективность оказания медицинской помощи.

Цель исследования — оценить организацию и качество амбулаторной медицинской помощи в ПМСП на основе опроса медицинских работников КГП на ПХВ «Больница города Приозерск» УЗКО.

Материалы и методы

Дизайн исследования

Проведено описательное поперечное исследование с использованием метода анкетирования.

Место и период проведения

Исследование проводилось на базе КГП на ПХВ «Больница города Приозерск» УЗКО (Республика Казахстан). Сбор данных осуществлялся в период с 27 апреля по 15 мая 2025 года.

Участники исследования

В исследовании приняли участие 60 сотрудников, работающих в подразделениях первичной медико-санитарной помощи. Включение в исследование осуществлялось методом сплошной выборки среди всех сотрудников ПМСП, давших добровольное информированное согласие на участие.

Критерии включения

В исследование включались сотрудники подразделений ПМСП КГП на ПХВ «Больница города Приозерск» УЗКО, давшие добровольное информированное согласие на участие.

Критерии исключения

Из исследования исключались сотрудники, отказавшиеся от участия или предоставившие неполностью заполненные анкеты.

Инструменты исследования

Для сбора данных использовалась авторская анкета, разработанная в рамках магистерского проекта. Анкета включала 22 вопроса, сгруппированные в пять тематических разделов:

Раздел 1. Общая информация (социально-демографические характеристики).

Раздел 2. Организация амбулаторной помощи.

Раздел 3. Ресурсное обеспечение.

Раздел 4. Оценка качества медицинской помощи.

Раздел 5. Перспективы совершенствования.

Вопросы были представлены в формате закрытых и полужакрытых вариантов ответов, а также со множественным выбором ответов.

Этические аспекты

Исследование одобрено локальным этическим комитетом. Участие в опросе являлось добровольным и анонимным. Все респонденты были проинформированы о целях исследования и дали согласие на обработку данных.

Статистический анализ

Полученные данные обработаны методами описательной статистики. Результаты представлены в виде абсолютных частот и относительных показателей (%).

Результаты

Социально-демографическая характеристика респондентов

В исследовании приняли участие 60 сотрудников ПМСП: женщины — 55 человек (91,7 %), мужчины — 5 человек (8,3 %).

По структуре должностей доминирующей группой явились медицинские сестры — 42 человека (70,0 %). Остальные участники были представлены врачами ПМСП, фельдшерами, акушерками, административными сотрудниками, прививочными медсестрами и хирургом.

Типичным участником исследования являлась женщина — медицинская сестра подразделения первичной медико-санитарной помощи КГП на ПХВ «Больница города Приозерск» УЗКО, принимающая участие в оказании амбулаторной медицинской помощи населению.

Оценка организации амбулаторной службы

При анализе организации работы амбулаторной службы большинство респондентов дали положительную оценку: «хорошо» — 33,3 % ($n = 20$), «отлично» — 25,0 % ($n = 15$). Вместе с тем значительная доля сотрудников — 38,3 % ($n = 23$) — оценила организацию работы как «удовлетворительно», что свидетельствует о наличии резервов для улучшения (таблица 1).

Таблица 1. Оценка организации работы амбулаторной службы

Оценка	n	%
Отлично	15	25,0
Хорошо	20	33,3
Удовлетворительно	23	38,3
Неудовлетворительно	2	3,4
Итого	60	100

Достаточность времени для качественного приема пациентов

Большинство респондентов — 66,7 % ($n = 40$) — указали, что им достаточно времени для качественного приема пациентов. Частично достаточным время считают 18,3 % ($n = 11$), тогда как 15,0 % ($n = 9$) отметили нехватку времени (таблица 2).

Таблица 2. Достаточность времени для приема пациентов

Ответ	n	%
Да	40	66,7
Частично	11	18,3
Нет	9	15,0
Итого	60	100

Факторы, негативно влияющие на качество медицинской помощи

Респондентам было предложено отметить все значимые факторы, негативно влияющие на качество медицинской помощи (вопрос со множественным выбором, $n = 59$). Наиболее часто называемым негативным фактором стало недостаточное финансирование — 30 ответов (50,8 %). На втором месте с одинаковой частотой указывались кадровый дефицит и высокая рабочая нагрузка — по 29 ответов (49,2 %). Далее следуют недостаточное обеспечение оборудованием — 22 (37,3 %), низкая мотивация персонала — 14 (23,7 %), избыточная документация и низкая активность населения — по 13 ответов (22,0 %). Наименее значимым фактором была названа недостаточная цифровизация — 4 ответа (6,8 %) (таблица 3).

Таблица 3. Факторы, негативно влияющие на качество медицинской помощи

Фактор	n	%
Недостаточное финансирование	30	50,8
Кадровый дефицит	29	49,2
Высокая нагрузка	29	49,2
Недостаточное обеспечение оборудованием	22	37,3
Низкая мотивация персонала	14	23,7
Избыточная документация	13	22,0
Низкая активность населения	13	22,0
Недостаточная цифровизация	4	6,8

Примечание. Вопрос со множественным выбором, в связи с чем сумма процентов превышает 100 %.

Меры по улучшению качества медицинской помощи

При оценке приоритетных мер по улучшению качества медицинской помощи (вопрос со множественным выбором, $n = 60$) наибольшее число респондентов указало на необходимость повышения заработной платы — 44 человека (73,3 %). Увеличение кадрового состава считают необходимым 33 респондента (55,0 %), повышение квалификации персонала — 27 (45,0 %). Снижение рабочей нагрузки отметили 22 участника (36,7 %), улучшение оснащения — 21 (35,0 %), развитие цифровизации — 17 (28,3 %), усиление профилактической работы — 13 (21,7 %), совершенствование маршрутизации пациентов — 7 (11,7 %), иные меры — 3 (5,0 %) (таблица 4).

Таблица 4. Приоритетные меры по улучшению качества медицинской помощи

Мера	n	%
Повышение заработной платы	44	73,3
Увеличение кадров	33	55,0
Повышение квалификации персонала	27	45,0
Снижение нагрузки	22	36,7
Улучшение оснащения	21	35,0
Развитие цифровизации	17	28,3
Усиление профилактической работы	13	21,7
Совершенствование маршрутизации	7	11,7
Другое	3	5,0

Примечание. Вопрос со множественным выбором, в связи с чем сумма процентов превышает 100 %.

Обсуждение

Полученные результаты отражают актуальные проблемы функционирования ПМСП и согласуются с данными ряда отечественных и зарубежных исследований.

Преобладание женщин среди респондентов (91,7 %) соответствует общей тенденции феминизации медицинских профессий среднего звена в Казахстане и странах постсоветского пространства [1]. Доминирование медицинских сестер в структуре персонала (70,0 %) подчеркивает ключевую роль сестринского звена в организации ПМСП.

Позитивным результатом является то, что большинство сотрудников (58,3 %) оценивают организацию амбулаторной службы как «хорошо» или «отлично». Вместе с тем каждый третий респондент (38,3 %) дал лишь удовлетворительную оценку, что указывает на существенные резервы для совершенствования системы организации помощи.

Примечательно, что 66,7 % участников отметили достаточность времени для приема пациентов. Однако совокупно 33,3 % сотрудников испытывают дефицит времени — полный или частичный, что непосредственно сказывается на качестве медицинской помощи и согласуется с данными о высокой нагрузке на врачей ПМСП [7].

Среди факторов, негативно влияющих на качество помощи, лидирующие позиции занимают недостаточное финансирование (50,8 %), кадровый дефицит (49,2 %) и высокая рабочая нагрузка (49,2 %). Данная триада проблем является системной и характерна для ПМСП в странах с переходной экономикой [5]. Недостаточное обеспечение оборудованием, отмеченное 37,3 % респондентов, свидетельствует о необходимости целенаправленных инвестиций в материально-техническую базу первичного звена.

Полученные результаты согласуются с данными Всемирной организации здравоохранения, согласно которым кадровый дефицит и высокая нагрузка являются одними из основных проблем ПМСП [6; 8; 9]. Аналогичные тенденции отмечаются в странах СНГ и сопровождаются риском профессионального выгорания медицинских работников [6]. Высокая нагрузка и дефицит времени могут снижать качество медицинской помощи и способствовать оттоку кадров из первичного звена здравоохранения.

Доминирование медицинских сестер в структуре респондентов (70,0 %) подчеркивает важную роль сестринского персонала в обеспечении доступности и непрерывности амбулаторной помощи [10; 11]. Международные исследования рассматривают укрепление кадрового потенциала сестринского персонала как один из ключевых факторов повышения эффективности ПМСП [2; 12].

В качестве приоритетной меры улучшения функционирования ПМСП подавляющее большинство респондентов (73,3 %) назвали повышение заработной платы, что является значимым индикатором неудовлетворенности уровнем материального вознаграждения и может служить фактором оттока кадров из ПМСП. Высокий запрос на увеличение кадрового состава (55,0 %) и повышение квалификации (45,0 %) подтверждает необходимость комплексного подхода к кадровой политике в первичном звене здравоохранения.

Ограничения исследования

Настоящее исследование имеет ряд ограничений. Во-первых, оно проводилось на базе одной медицинской организации, что ограничивает возможность обобщения результатов. Во-вторых, анкетный метод предполагает субъективность оценок. В-третьих, относительно небольшой объем выборки ($n = 60$) снижает статистическую мощность исследования.

Заключение

Проведенное исследование позволило выявить ключевые проблемы в организации ПМСП с позиции медицинских работников КПП на ПХВ «Больница города Приозерск» УЗКО. Несмотря на преимущественно положительную оценку организации амбулаторной службы, значительная часть персонала указывает на системные барьеры: недостаточное финансирование, кадровый дефицит и высокую рабочую нагрузку. Приоритетными мерами совершенствования деятельности ПМСП, по мнению сотрудников, являются повышение заработной платы, увеличение кадрового состава и повышение квалификации персонала.

Полученные данные могут служить основой для разработки управленческих решений, направленных на повышение качества и доступности первичной медико-санитарной помощи на уровне медицинской организации и региона в целом.

Результаты исследования подчеркивают необходимость дальнейшего укрепления кадрового потенциала и совершенствования организационных механизмов ПМСП в условиях реформирования системы здравоохранения Республики Казахстан.

Литература:

1. Primary Health Care: Closing the Gap Between What We Know and What We Do / World Health Organization. — Geneva : WHO, 2023.
2. Руководство по организации первичной медико-санитарной помощи в амбулаторных условиях / Кулкаева Г. У., Темкова З. М., Омирбаева Б. С., Серикбаев Н. С., Хасенова А. Ж. — Астана : ННЦРЗ МЗ РК, 2023. — ISBN 978-601-305-588-6.
3. Государственная программа развития здравоохранения «Саламатты Қазақстан» / Министерство здравоохранения Республики Казахстан. — Астана : [б. и.], 2020.
4. Денисов, И. Н. Общая врачебная практика : национальное руководство / И. Н. Денисов. — М. : ГЭОТАР-Медиа, 2020.
5. Bodenheimer, T. From Triple to Quadruple Aim: Care of the Patient Requires Care of the Provider / T. Bodenheimer, C. Sinsky // The Annals of Family Medicine. — 2014. — Vol. 12, iss. 6. — P. 573–576.
6. Rechel, B. Facets of Public Health in Europe / B. Rechel, M. McKee. — Berkshire : Open University Press, 2014.
7. Global Strategic Directions for Nursing and Midwifery 2021–2025 / World Health Organization. — Geneva : WHO, 2021.
8. Starfield, B. Primary Care: Balancing Health Needs, Services, and Technology / B. Starfield. — New York : Oxford University Press, 1998.
9. Health at a Glance 2023: OECD Indicators / OECD. — Paris : OECD Publishing, 2023.
10. Старшинин, А. В. Роль медицинской сестры в мультидисциплинарной команде при оказании первичной медико-санитарной помощи: обзор литературы / А. В. Старшинин, Н. Н. Камынина, А. С. Тимофеева // Здоровье мегаполиса. — 2024. — Т. 5, № 4 (1). — С. 131–141. — DOI: 10.47619/2713–2617.zm.2024.v5i4p1;131–141
11. Маханбеткулова, Д. Н. Медсестра расширенной практики: текущее состояние и перспективы развития / Д. Н. Маханбеткулова // Фтизиопульмонология. — 2024. — № 1 (43). — С. 106–117.
12. Роль медицинской сестры в оказании медицинской помощи пациентам пожилого возраста / Б. К. Исенова, Г. Е. Аимбетова, М. Kanushina [и др.] // Фармация Казахстана. — 2023. — № 3 (248). — С. 187–192.

Нестероидные противовоспалительные препараты в хирургии: баланс между анальгезией и риском периоперационных осложнений

Чкалин Иван Витальевич, студент

Научный руководитель: Нузова Ольга Борисовна, доктор медицинских наук, профессор;

Научный руководитель: Белянин Виталий Васильевич, кандидат медицинских наук, доцент

Оренбургский государственный медицинский университет

В статье проведён систематический анализ периоперационного применения нестероидных противовоспалительных препаратов в хирургии. Рассмотрены механизмы анальгезии, роль в мультимодальной анальгезии и протоколах уско-

ренного восстановления после операции (Enhanced Recovery After Surgery, ERAS). Систематизированы данные о трёх основных рисках: кровотечении, остром почечном повреждении (ОПП) и несостоятельности анастомозов (Anastomotic Leakage, AL). Выявлены противоречия доказательной базы, предложен дифференцированный подход к выбору НПВП и алгоритмы минимизации осложнений.

Ключевые слова: НПВП, хирургия, мультимодальная анальгезия, кровотечение, нефротоксичность, ОПП, несостоятельность анастомоза

Введение

Послеоперационная боль остаётся серьёзной проблемой, влияющей на течение раннего периода, сроки госпитализации и риск хронического болевого синдрома. Традиционное применение опиоидов сопряжено с угнетением дыхания, тошнотой, запором и зависимостью [8, с. 1–7]. Мультимодальная анальгезия, сочетающая препараты разных механизмов, позволяет усилить эффект и снизить дозу каждого компонента [6]. НПВП занимают центральное место в таких схемах благодаря мощному обезболивающему и противовоспалительному действию, уменьшая потребность в опиоидах на 30–50 % [2, с. 294–296]. Они являются обязательным элементом протоколов ускоренного восстановления после операции (Enhanced Recovery After Surgery, ERAS) [6]. Однако безопасность НПВП вызывает дискуссии из-за рисков кровоточивости, нефротоксичности и несостоятельности анастомозов [8, с. 1–7]. Разные препараты класса существенно различаются по профилю безопасности, что требует дифференцированного подхода.

Цель настоящей работы — систематический анализ данных по безопасности НПВП при абдоминальных операциях, выявление баланса эффективности/риск и выработка практических рекомендаций.

Фармакологические основы действия НПВП, механизм действия и классификация. Основной механизм — ингибирование циклооксигеназы (ЦОГ), что снижает синтез простагландинов, простаглицлинов и тромбоксана [2, с. 294–296]. Выделяют ЦОГ1 (физиологические функции) и ЦОГ2 (индуцируется при воспалении). По селективности различают:

1. **Неселективные ингибиторы ЦОГ** (диклофенак, ибупрофен, кеторолак, напроксен): блокируют как ЦОГ1, так и ЦОГ2, что обуславливает как высокую анальгетическую активность, так и широкий спектр побочных эффектов.

2. **Селективные ингибиторы ЦОГ2** (целекоксиб, эторикоксиб, парекоксиб, валдекоксиб): преимущественно блокируют изоформу ЦОГ2, что позволяет сохранить синтез простагландинов, защищающих слизистую желудка и поддерживающих почечный кровоток, однако может сопровождаться повышением тромбообразования.

3. **Препараты с преимущественно центральным анальгетическим действием** (метамизол натрия, парацетамол): механизм действия лишь частично связан с ингибированием ЦОГ и включает также воздействие на центральные звенья ноцицептивной системы.

Снижение простагландинов в очаге воспаления даёт лечебный эффект, а блокада физиологических простагландинов лежит в основе нежелательных явлений [2, с. 294–296].

НПВП в протоколах ERAS. Протоколы ускоренного восстановления (Enhanced Recovery After Surgery, ERAS) представляют собой мультидисциплинарные программы для снижения стресс-ответа и ускорения восстановления [6]. НПВП в их составе позволяют снизить потребление опиоидов и частоту их побочных эффектов, уменьшить воспаление, ускорить восстановление моторики ЖКТ и активизацию пациентов, сократить сроки госпитализации [8, с. 1–7]. Мета-анализы подтверждают снижение боли, опиоидного потребления и ускорение восстановления кишечника [8, с. 1–7]. Однако преимущества должны быть взвешены против рисков, особенно у пациентов с отягощённым фоном.

Риск периоперационного кровотечения. Патфизиологические механизмы. НПВП ингибируют ЦОГ1 в тромбоцитах, снижая синтез тромбоксана A_2 и агрегацию тромбоцитов [2, с. 294–296]. Это обратимо. Селективные ЦОГ2 практически не влияют на тромбоциты. Клинически кровоточивость может проявляться увеличением кровопотери, гематом, потребностью в гемотранфузиях. В исследовании после резекции прямой кишки ($n=1663$) кровотечение чаще встречалось при приёме НПВП (4,0 % против 1,2 %, $p=0,003$), особенно неселективных и при многократном введении [4, с. 1270–1278]. Однако при соблюдении дозировок и отсутствии дополнительных факторов (антикоагулянты, нарушения гемостаза) клинически значимого увеличения кровоточивости может не быть [1, с. 904–910]. Риск зависит от типа препарата, дозы, длительности, операции и исходного гемостаза.

Нефротоксичность НПВП и риск острого почечного повреждения. Почки являются одним из органов-мишеней для побочных эффектов НПВП. Механизм — блокада почечных простагландинов, поддерживающих кровоток и фильтрацию, что ведёт к вазоконстрикции и снижению СКФ [2, с. 294–296; 7]. Особенно опасно при гиповолемии, кровопотере, сепсисе, сердечной недостаточности [7]. Факторы риска: пожилой возраст, ХБП, диабет, гипертензия, приём других нефротоксичных препаратов. ОПП — серьёзное осложнение, увеличивающее госпитализацию, потребность в диализе и летальность [7]. Частота после больших абдоминальных операций — 10–30 %, в ОРИТ — до 50 % [7]. Даже лёгкое ОПП повышает риск ХБП и сердечно-сосудистых событий.

В исследовании критических больных ($n=1157$) НПВП ассоциировались с более высокой частотой ОПП (67 % против 51 %, $p=0,009$), тяжёлого ОПП (34 % против 26 %, $p=0,008$), ОШ 1,98 [7]. Тяжёлое ОПП снижало 2летнюю выживаемость (скорр. ОР 1,42) [7]. По данным Cochrane, НПВП могут вызывать ОПП у 5 % пациентов, а при стрессе — чаще [7]. Однако при коротких курсах у пациентов без факторов риска достоверного увеличения почечных осложнений не выявлено [1, с. 904–910]. Ключевое значение имеют дополнительные факторы и гемодинамика, поэтому стратификация и мониторинг обязательны.

Несостоятельность анастомозов: патофизиологические механизмы возможной связи. Несостоятельность анастомозов (Anastomotic Leakage, AL) остаётся одним из наиболее грозных осложнений в абдоминальной хирургии, особенно после резекций желудка, тонкой и толстой кишки. Частота этого осложнения варьирует от 3 до 21 % в зависимости от локализации анастомоза, типа оперативного вмешательства и исходного состояния пациента, и ассоциирована с высокой летальностью (до 30–40 %) и значительным ухудшением отдалённых онкологических результатов [6, 7].

Потенциальные механизмы связи с НПВП: нарушение микроциркуляции, ингибирование синтеза коллагена, подавление воспалительного ответа, тромботические микроангиопатии [2, с. 294–296; 3, с. 12–19]. Доказательная база гетерогенна. Umbrella review 2026 года (27 обзоров) показал единодушие в эффективности НПВП, но данные по AL (13 обзоров) дискордантны; наилучшие данные указывают на риск для неселективных НПВП, для селективных ЦОГ2 убедительных доказательств нет; качество обзоров — критически низкое [8, с. 1–7]. Напротив, ретроспективное когортное исследование 2026 года ($n=18\,312$) выявило повышение риска AL именно для селек-

тивных ЦОГ2 (ОШ 1,95, $p<0,001$), а для неселективных связи не было [5]. Исследование после проктэктомии ($n=1663$) показало, что раннее (первые сутки) назначение НПВП ассоциировано с ранней AL (до 6 суток): 8,4 % против 4,6 % ($p=0,041$), преимущественно за счёт неселективных (ОШ 1,717) и многократных введений (ОШ 1,687), но не с поздней AL [4, с. 1270–1278]. Повышенная чувствительность у мужчин (ОШ 1,836) и без стом (ОШ 1,689) [4, с. 1270–1278]. Обзор 2026 года подтвердил связь НПВП с AL, а также факторы ASA 3–4, мужской пол, диабет, гипертензия, ХБП [6]. Многоцентровое проспективное исследование ($n=4164$) не выявило различий в частоте AL (5,4 % vs 4,6 %, $p=0,349$) и ОПП (14,3 % vs 13,8 %, $p=0,666$) [1, с. 904–910]. Таким образом, связь сложная и зависит от типа препарата, режима, сроков, локализации, пола и стомы.

Частота периоперационных осложнений. В структуре послеоперационных осложнений у пациентов высокого риска при плановой абдоминальной хирургии были выявлены наиболее значимые неблагоприятные исходы, которые отличались по частоте и клинической значимости. Наибольшую долю среди осложнений составил парез кишечника, что указывает на ведущую роль ранних функциональных нарушений в послеоперационном периоде. Частота осложнений распределялась следующим образом (рис. 1): парез кишечника 4,8 %, послеоперационное кровотечение 2,6 %, пневмония 2,4 %, раневая инфекция 2,4 %, несостоятельность анастомоза 2,1 %, острое повреждение почек 1,9 %, послеоперационный делирий 1,9 %. Это свидетельствует о том, что в раннем послеоперационном периоде у пациентов высокого риска доминируют осложнения, связанные с нарушением моторики желудочно-кишечного тракта, а также инфекционные и хирургические осложнения [11, с. 51–63].

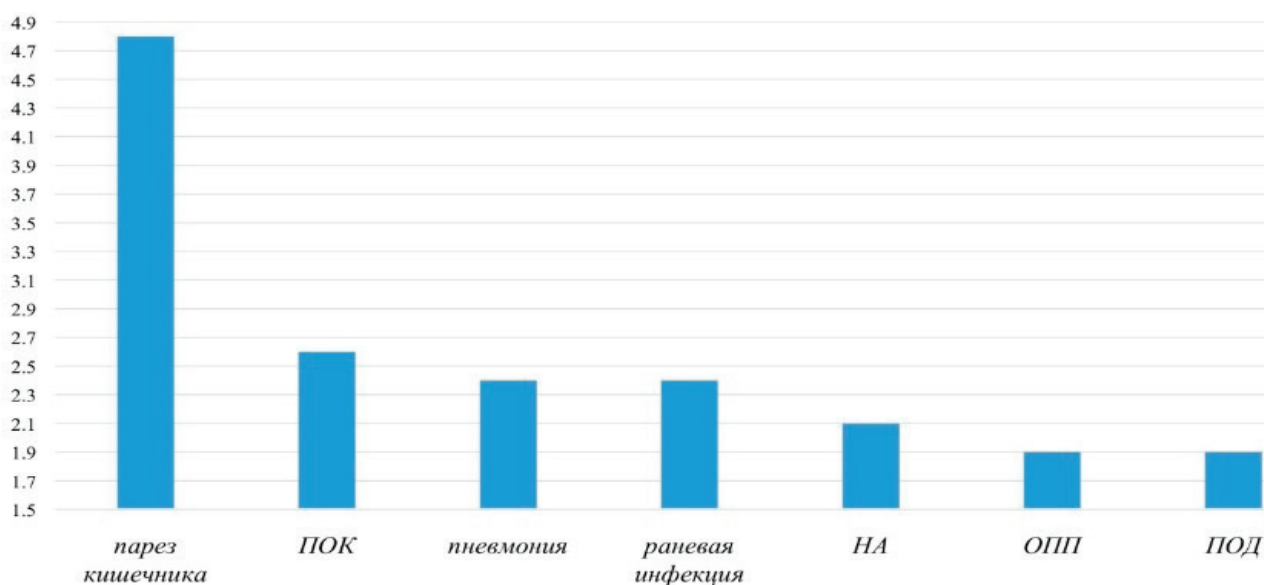


Рис. 1. Частота послеоперационных осложнений у пациентов высокого риска при плановой абдоминальной хирургии [11, с. 51–63]. ПОК — послеоперационное кровотечение, НА — несостоятельность анастомозов, ОПП — острое повреждение почек, ПОД — послеоперационный делирий.

Сравнительная характеристика различных групп НПВП

Таблица 1. Сравнительная характеристика эффективности и рисков различных групп НПВП в периоперационном периоде

Параметр	Неселективные ингибиторы ЦОГ (диклофенак, ибупрофен, кеторолак)	Селективные ингибиторы ЦОГ2 (целекоксиб, эторикоксиб, парекоксиб)	Метамизол натрия
Анальгетический эффект	Высокий	Высокий	Умеренный
Опиоидсберегающий эффект	Выраженный	Выраженный	Умеренный
Риск кровотечения	Умеренный (дозозависимый)	Низкий	Низкий
Риск нефротоксичности	Умеренный	Низкий	Низкий
Риск несостоятельности анастомоза	Возможен (данные противоречивы)	Возможен (по некоторым данным)	Не изучался
Риск желудочно-кишечных осложнений	Высокий	Низкий	Низкий
Кардиоваскулярный риск	Низкий (кроме высоких доз)	Повышен (при длительном применении)	Низкий

Примечание: данные основаны на результатах исследований последних лет и носят обобщающий характер; при индивидуальном подходе профиль риска может существенно варьировать.

Стратегии минимизации рисков и практические рекомендации. На основании проведённого анализа современной литературы можно предложить следующие подходы к безопасному и эффективному применению НПВП в периоперационном периоде.

1. **Тщательный отбор пациентов.** Перед назначением НПВП необходимо провести комплексную оценку факторов риска. **Функция почек:** определение уровня креатинина сыворотки, расчёт скорости клубочковой фильтрации (СКФ) с использованием формулы CKD-EPI или MDRD. При СКФ < 60 мл/мин/1,73 м² применение НПВП требует особой осторожности. **Гемостаз:** оценка количества тромбоцитов, протромбинового времени, активированного частичного тромбопластинового времени. При нарушениях гемостаза или тромбоцитопении НПВП противопоказаны. **Желудочно-кишечный тракт:** наличие язвенной болезни в анамнезе, желудочно-кишечных кровотечений, одновременный приём глюкокортикостероидов или антикоагулянтов. **Сердечно-сосудистая система:** наличие ишемической болезни сердца, сердечной недостаточности, неконтролируемой артериальной гипертензии (особенно важно для селективных ингибиторов ЦОГ2). **Возраст:** пожилые пациенты имеют более высокий риск всех НПВП-ассоциированных осложнений.

2. **Выбор препарата с учетом профиля безопасности.** При **высоком риске кровотечения** (приём антикоагулянтов, нарушения гемостаза, обширные оперативные вмешательства с высокой кровопотерей) предпочтение следует отдавать **селективным ингибиторам ЦОГ2** или метамизолу натрия. При **высоком риске несостоятельности анастомозов** (колоректальные операции, пациенты с факторами риска АЛ) данные литературы не позволяют сделать однозначный выбор в пользу какого-либо класса НПВП. Многие авторы рекомендуют **ограничивать при-**

менение неселективных НПВП в раннем послеоперационном периоде, особенно при многократном введении [8, с. 1–7; 4, с. 1270–1278]. При **хронической болезни почек** или **высоком риске ОПП** предпочтительны селективные ингибиторы ЦОГ2 или препараты с минимальным нефротоксическим потенциалом, однако и они требуют тщательного мониторинга. При **высоком кардиоваскулярном риске** следует избегать длительного применения селективных ингибиторов ЦОГ2.

3. **Минимизация дозы и длительности курса.** Следует использовать **минимальную эффективную дозу** и **максимально короткий курс** (как правило, не более 3–5 суток), особенно у пациентов с факторами риска. При необходимости более длительной анальгезии следует рассмотреть возможность перехода на препараты с другим механизмом действия или комбинированной терапии с использованием парацетамола, трамадола или других ненаркотических анальгетиков.

4. **Мониторинг осложнений.** В раннем послеоперационном периоде необходим регулярный контроль: **диуреза** (почасовой в первые сутки, затем суточный); **уровня креатинина сыворотки** (ежедневно или через день в зависимости от исходного уровня и динамики); **признаков желудочно-кишечного кровотечения** (снижение гемоглобина, мелена, гематемезис); **клинических симптомов несостоятельности анастомоза** (боли в животе, лихорадка, тахикардия, лейкоцитоз, перитонеальные симптомы, выделение содержимого кишечника по дренажам).

5. **Альтернативные стратегии анальгезии.** У пациентов с множественными факторами риска, у которых применение НПВП сопряжено с недопустимо высоким риском, следует рассматривать альтернативные стратегии мультимодальной анальгезии: регионарные методы анестезии (эпидуральная анальгезия, блокады нервов); па-

рацетамол (внутривенно или перорально); трамадол или другие слабые опиоиды; габапентиноиды (габапентин, прегабалин): при нейропатическом компоненте боли; кетамин в субдиссоциативных дозах; местные анесте-

тики в ране (инфильтрация, катетеры для непрерывной инфузии).

На рисунке 2 представлен алгоритм принятия решения о применении НПВП в периоперационном периоде.

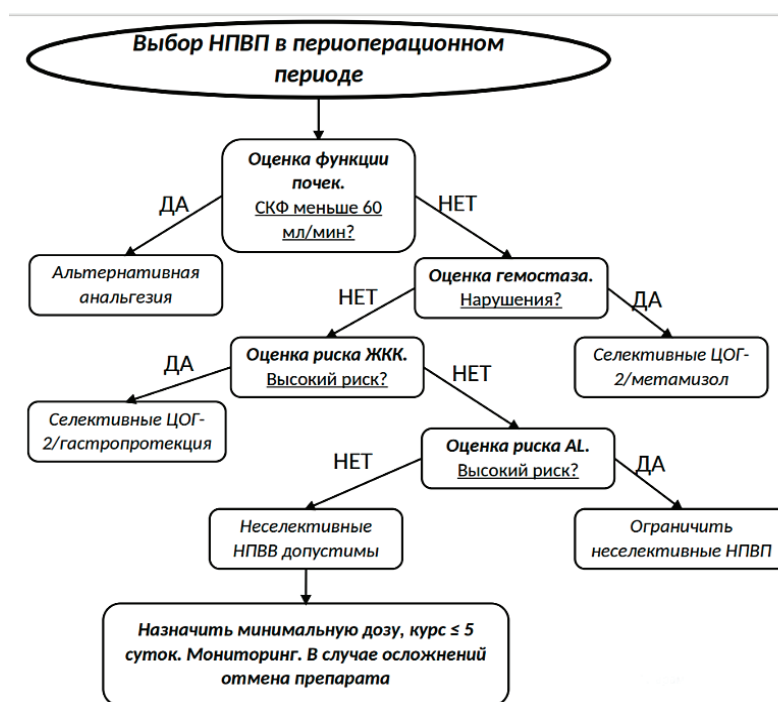


Рис. 2. Алгоритм выбора НПВП в периоперационном периоде с учётом индивидуальных факторов риска (предложен авторами на основании анализа литературы). НПВП — нестероидные противовоспалительные препараты, СКФ — скорость клубочковой фильтрации, АЛ — несостоятельность анастомозов

Заключение

НПВП являются эффективным компонентом мультимодальной анальгезии и ERAS, обеспечивающий обезболивание, опиоидсбережение и ускорение восстановления. Однако их применение требует дифференцированного подхода с учётом рисков кровотечения, ОПП и АЛ. Докладательная база противоречива из-за гетерогенности

исследований, различий в препаратах, режимах и популяциях. Особенно дискуссионна связь с АЛ — данные варьируют от отсутствия до значительного повышения, причём разные исследования указывают на разные группы НПВП. Необходимы дальнейшие проспективные рандомизированные исследования с чёткой стратификацией. Перспективно изучение молекулярных механизмов влияния НПВП на заживление анастомозов.

Литература:

1. Perioperative non-steroidal anti-inflammatory drugs (NSAID) administration and acute kidney injury (AKI) in major gastrointestinal surgery: A prospective, multicenter, propensity matched cohort study // *Annals of Surgery*. — 2022. — Vol. 275, No. 5. — P. 904–910. DOI: 10.1097/SLA.0000000000004314.
2. Joshi G. P., et al. Nonsteroidal anti-inflammatory drugs in the perioperative period: current controversies and concerns // *British Journal of Anaesthesia*. — 2025. — Vol. 134, No. 2. — P. 294–296. DOI: 10.1016/j.bja.2024.10.018.
3. Kowal M. R., Nicholls A., Jayne D. G., Chapman S. J. Non-Steroidal Anti-Inflammatory Drugs After Abdominal Surgery: An Umbrella Review of Existing Evidence // *ANZ Journal of Surgery*. — 2026. — Vol. 96, No. 1. — P. 12–19. DOI: 10.1111/ans.70468.
4. Aviran E., Assaf D., Zaghiyan K. N., Fleshner P. Non-steroidal Anti-inflammatory Drugs and Timing-specific Anastomotic Leak Risk Following Minimally Invasive Proctectomy: A Retrospective Cohort Study // *Annals of Surgical Oncology*. — 2026. — Vol. 33. — P. 1270–1278. DOI: 10.1245/s10434-025-18715-6.
5. Perioperative nonsteroidal anti-inflammatory drugs and the risk for anastomotic leakage in patients undergoing major gastrointestinal surgery: a retrospective cohort study // *BMC Anesthesiology*. — 2026. — Vol. 26. — Art. no. 59. DOI: 10.1186/s12871-025-03560-7.

6. Risk and protective factors for postoperative anastomotic leakage in esophageal and gastrointestinal surgery: an umbrella review of meta-analyses and systematic reviews // *International Journal of Surgery*. — 2025. DOI: 10.1097/JS9.0000000000003308.
7. Incidence of kidney toxicity of non-steroidal anti-inflammatory drugs in critically ill patients // *Journal of Critical Care*. — 2025. DOI: 10.1016/j.jcrc.2024.154912.
8. Effect of Perioperative Dexamethasone and Different NSAIDs on Anastomotic Leak Risk: A Propensity Score Analysis // *Annals of Surgery*. — 2017. — Vol. 266, No. 1. — P. 1–7. DOI: 10.1097/SLA0000000000002013.
9. Postoperative non-steroidal anti-inflammatory drugs and anastomotic leakage in gastrointestinal surgery: a systematic review and meta-analysis // *World Journal of Surgery*. 2020. DOI: 10.1007/s00268–020–05408–7.
10. Non-steroidal anti-inflammatory agents and anastomotic leak rates across colorectal cancer operations and anastomotic sites: a systematic review and meta-analysis of anastomosis-specific leak rate and confounding factors // *Diseases of the Colon & Rectum*. 2021. DOI: 10.1097/DCR.0000000000002185.
11. Эпидемиология неблагоприятных исходов в плановой абдоминальной хирургии: результаты проспективного наблюдательного многоцентрового исследования // *Вестник интенсивной терапии имени А. И. Салтанова*. — 2025. — № 3. — С. 51–63.

Молодой ученый

Международный научный журнал
№ 29 (632) / 2026

Выпускающий редактор Г. А. Письменная
Ответственные редакторы Е. И. Осянина, О. А. Шульга, З. А. Огурцова
Художник Е. А. Шишков
Подготовка оригинал-макета П. Я. Бурьянов, М. В. Голубцов, О. В. Майер

За достоверность сведений, изложенных в статьях, ответственность несут авторы.
Мнение редакции может не совпадать с мнением авторов материалов.
При перепечатке ссылка на журнал обязательна.
Материалы публикуются в авторской редакции.

Журнал размещается и индексируется на портале eLIBRARY.RU, на момент выхода номера в свет журнал не входит в РИНЦ.

Свидетельство о регистрации СМИ ПИ № ФС77-38059 от 11 ноября 2009 г., выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор).

ISSN-L 2072-0297

ISSN 2077-8295 (Online)

Учредитель и издатель: ООО «Издательство Молодой ученый». 420029, Республика Татарстан, г. Казань, ул. Академика Кирпичникова, д. 25, пом. 1, 3, 4, 5, 6.

Номер подписан в печать 29.07.2026. Дата выхода в свет: 05.08.2026.

Формат 60×90/8. Тираж 500 экз. Цена свободная.

Почтовый адрес редакции: 420140, Республика Татарстан, г. Казань, ул. Юлиуса Фучика, д. 94А, а/я 121.

Фактический адрес редакции: 420029, Республика Татарстан, г. Казань, ул. Академика Кирпичникова, д. 25, пом. 1, 3, 4, 5, 6.

E-mail: info@moluch.ru; <https://moluch.ru/>

Отпечатано в типографии издательства «Молодой ученый», 420029, Республика Татарстан, г. Казань, ул. Академика Кирпичникова, д. 25, пом. 1, 3, 4, 5, 6.