

**Событийно-ориентированная архитектура
в цифровом образовании:
педагогические возможности и ограничения**
**Event-driven architecture in digital education:
pedagogical opportunities and limitations**

Авторы статьи

Закриев Исафил Рустамович,
студент ФГБОУ ВО «Грозненский государственный нефтяной
технический университет имени академика
М. Д. Миллионщикова», г. Грозный, Российская Федерация
zakrievhakim01@gmail.com
ORCID: 0009-0002-1970-1280

Боприев Дени Абубакарович,
студент ФГБОУ ВО «Грозненский государственный нефтяной
технический университет имени академика
М. Д. Миллионщикова», г. Грозный, Российская Федерация
boprievdeni@gmail.com
ORCID: 0009-0008-8914-9165

Конфликт интересов

Конфликт интересов не указан.

Для цитирования

Закриев И. Р., Боприев Д. А. Событийно-ориентированная архитектура в цифровом образовании: педагогические возможности и ограничения // Научно-методический электронный журнал «Концепт». – 2026. – № 08. – С. 316–331. – URL: <https://e-koncept.ru/2026/261226.htm> – DOI: 10.24412/2304-120X-2026-11226

Authors of the article

Israfil R. Zakriev,
Student, Grozny State Petroleum Technical University
named after Academician M. D. Millionshchikov, Grozny,
Russian Federation
zakrievhakim01@gmail.com
ORCID: 0009-0002-1970-1280

Deni A. Bopriev,
Student, Grozny State Petroleum Technical University
named after Academician M. D. Millionshchikov, Grozny,
Russian Federation
boprievdeni@gmail.com
ORCID: 0009-0008-8914-9165

Conflict of interest statement

Conflict of interest is not declared

For citation

I. R. Zakriev, D. A. Bopriev, Event-driven architecture in digital education: pedagogical opportunities and limitations // Scientific-methodological electronic journal "Koncept". – 2026. – No. 08. – P. 316–331. – URL: <https://e-koncept.ru/2026/261226.htm> – DOI: 10.24412/2304-120X-2026-11226

Поступила в редакцию <i>Received</i>	24.05.26	Получена положительная рецензия <i>Received a positive review</i>	15.07.26
Принята к публикации <i>Accepted for publication</i>	15.07.26	Опубликована <i>Published</i>	31.08.26



Аннотация

Цифровая трансформация высшего образования сопровождается ростом требований к устойчивости, масштабируемости и адаптивности образовательных платформ. Традиционные синхронные клиент-серверные архитектуры, основанные на последовательном обмене запросами и ответами, все чаще демонстрируют ограничения в условиях высокой нагрузки, массового удаленного доступа и необходимости оперативной обработки учебных событий. В статье рассматривается событийно-ориентированная архитектура как инструмент повышения эффективности цифровой образовательной среды. Актуальность исследования обусловлена необходимостью выявления не только технических, но и педагогических эффектов внедрения событийной модели в образовательные системы. Цель исследования заключается в выявлении и систематизации педагогических эффектов, возникающих при внедрении событийно-ориентированной архитектуры в цифровые образовательные платформы. Методологическую основу работы составили системный, деятельностный и компетентностный подходы. В исследовании использовались методы сравнительного анализа архитектурных паттернов, моделирования типовых учебных сценариев, контент-анализа научных публикаций, а также элементы эмпирического наблюдения в рамках пилотного проекта, реализованного в Грозненском государственном нефтяном техническом университете с использованием брокера сообщений RabbitMQ. В ходе исследования установлено, что применение событийно-ориентированной архитектуры способствует сокращению времени отклика образовательной платформы, снижению нагрузки на преподавателей за счет автоматизации рутинных операций, расширению возможностей персонализации образовательных траекторий и накоплению полного цифрового следа учебной активности. В пилотной эксплуатации зафиксировано сокращение времени первичного отклика системы в полтора раза и двукратное снижение числа обращений в службу технической поддержки из-за проблем с зависанием интерфейса. Предложена авторская классификация педагогических эффектов, включающая дидактические, мотивационные, адаптационные, аналитические и организационные эффекты. Одновременно выявлены и систематизированы ограничения внедрения событийной модели, включая дублирование событий, нарушение порядка обработки сообщений, проблему согласованности данных в конечном счете и возросшую сложность отладки распределенных систем. Теоретическая значимость исследования состоит в развитии междисциплинарного подхода к анализу цифровой образовательной среды на стыке педагогики и программной архитектуры. Практическая значимость заключается в возможности использования результатов исследования при модернизации систем управления обучением, проектировании систем аналитики обучения и внедрении адаптивных образовательных сервисов в вузах.

Ключевые слова

событийно-ориентированная архитектура, цифровое образование, event-driven architecture, RabbitMQ, LMS, learning analytics, образовательные платформы, асинхронные системы

Благодарности

Авторы выражают благодарность Наталье Анатольевне Моисеенко, кандидату педагогических наук, профессору кафедры «Информационные технологии» Грозненского государственного нефтяного технического университета имени академика М. Д. Миллионщикова, за ценные советы и помощь при подготовке статьи.

Abstract

The digital transformation of higher education is accompanied by growing requirements for the resilience, scalability, and adaptability of educational platforms. Traditional synchronous client-server architectures based on sequential request-response exchange increasingly demonstrate limitations under conditions of high load, mass remote access, and the need for operational processing of learning events. The article considers event-driven architecture as a tool for improving the efficiency of the digital educational environment. The relevance of the study is due to the need to identify not only technical but also pedagogical effects of implementing an event-driven model in educational systems. The aim of the study is to identify and systematize the pedagogical effects arising from the implementation of event-driven architecture in digital educational platforms. The methodological basis of the work includes systemic, activity-based and competency-based approaches. The study uses methods of comparative analysis of architectural patterns, modeling of typical educational scenarios, content analysis of scientific publications, as well as elements of empirical observation within a pilot project implemented at Grozny State Petroleum Technical University using the RabbitMQ message broker. The study found that the use of event-driven architecture helped to reduce the response time of the educational platform, reduce the load on teachers by automating routine operations, expand the possibilities of personalizing educational trajectories, and accumulate a complete digital footprint of learning activity. In pilot operation, a 1.5-fold reduction in the system's primary response time and a twofold reduction in the number of requests to the technical support service due to interface freezes were recorded. The authors propose a classification of pedagogical effects, including didactic, motivational, adaptive, analytical and organizational effects. At the same time, the limitations of implementing the event-driven model are identified and systematized, including event duplication, violation of the order of message processing, the problem of eventual consistency, and the increased complexity of debugging distributed systems. The theoretical significance of the study lies in the development of an interdisciplinary approach to the analysis of the digital educational environment at the intersection of pedagogy and software architecture. The practical significance lies in the possibility of using the research results in the modernization of learning management systems, the design of learning analytics systems, and the implementation of adaptive educational services in universities.

Key words

event-driven architecture, digital education, RabbitMQ, LMS, learning analytics, educational platforms, asynchronous systems

Acknowledgements

The authors express their gratitude to Natalia A. Moiseenko, Candidate of Pedagogical Sciences, Professor of the Department of Information Technologies at the Grozny State Petroleum Technical University named after Academician M.D. Millionshchikov for her valuable advice and assistance in preparing this article.

Введение / Introduction

Современную цифровизацию образования уже не представишь без глубокого прорастания технологий во все образовательные процессы – от плана занятия до итоговой оценки, как отмечают М. Е. Вайндорф-Сысоева и М. Л. Субочева [1]. Внедрение информационно-коммуникационных технологий перестало быть факультативным и превратилось в необходимое условие эффективной работы образовательных учреждений, по утверждению И. В. Роберт [2]. Цифровые платформы, LMS, мобильные приложения и интерактивные среды давно перестали быть просто удобной подпоркой и стали инфраструктурой, от которой напрямую зависит качество педагогической работы. Типичная ситуация: преподаватель проверяет задание, студент завершает тест в последние секунды, и интерфейс перестает отвечать. В такие моменты сложно не обвинить само приложение, но корень проблемы почти всегда в другом – он в способе организации взаимодействия между его компонентами. Именно поэтому качество программной инфраструктуры, обеспечивающей это взаимодействие, приобретает сегодня критическое значение.

Долгое время образовательные платформы уверенно держались на синхронной архитектуре REST (Representational State Transfer), как описано у К. Ричардсона [3]. Понятные цепочки «запрос – ответ» работали безотказно, пока нагрузка оставалась стабильной. Но ситуация изменилась: резко выросла доля мобильного доступа, количество одновременно подключенных устройств исчисляется десятками тысяч, а учебные сценарии требуют мгновенной реакции: автоматической проверки работ, фоновых уведомлений, адаптивной подстройки траектории, что подчеркивает М. Клеппманн [4]. В таких условиях синхронная модель начинает «спотыкаться». Поток последовательных вызовов накапливает задержки, словно снежный ком, жесткая связность компонентов снижает отказоустойчивость, а пиковые нагрузки во время сессии нередко оборачиваются деградацией сервиса. Преподаватели и их студенты все чаще сталкиваются с ситуацией, когда платформа «тормозит» именно в те моменты, когда от нее ждут максимальной оперативности, как отмечает С. Ньюман [5].

В противовес синхронному подходу событийно-ориентированная архитектура (EDA) предлагает принципиально иную логику, как показали Г. Хоп и Б. Вульф [6]. Здесь действие студента: отправка работы, прохождение теста, вход в курс – регистрируется как событие и немедленно подхватывается независимыми сервисами без блокировки интерфейса. Ключевые особенности: асинхронная обработка, слабая связность источников и потребителей, возможность накапливать полный журнал событий (event store) и доставлять одно событие множеству подписчиков через брокера сообщений, согласно Э. Таненбауму и М. ван Стину [7]. В результате система не заставляет пользователя ждать завершения всей цепочки операций, а сразу подтверждает прием и позже уведомляет о результате. Это не только устраняет субъективное восприятие задержек, но и существенно повышает устойчивость платформы к пиковым нагрузкам.

Промышленная разработка активно осваивает EDA уже полтора десятилетия – достаточно назвать Apache Kafka, RabbitMQ, облачные сервисы AWS и Azure, как пишут Н. Наркхеде с соавторами [8], и в последние годы этот интерес переместился на образовательный сектор, о чем свидетельствуют Р. Йонгелинг и М. Линдхольм [9]. Зарубежные университеты и EdTech-компании, такие как edX и Coursera, начали переводить аналитические конвейеры и системы уведомлений на событийную шину, как описывают А. Кусеф с соавторами [10]. В отечественной практике таких примеров

пока меньше, однако появляются экспериментальные проекты, например у А. В. Шевчука [11]. Однако если технические аспекты EDA освещены в литературе довольно подробно: сравнительные тесты производительности, инженерия надежности, архитектурные паттерны, как показано у Р. Гупты и П. Сингха [12], то вопрос о том, как именно переход на события отражается на педагогическом процессе, остается открытым, отмечают Д. Гашевич с соавторами [13]. Зарубежные исследователи фиксируют рост пропускной способности и расширение аналитических возможностей, по данным Р. Фергюсона [14], но не формулируют, каким образом эти изменения сказываются на мотивации студента, качестве обратной связи, загрузке преподавателя или объективности оценивания. Отечественная же педагогическая наука эту тему практически не затрагивает, как констатирует С. В. Титова [15].

Таким образом, сложился разрыв: образовательные учреждения начинают внедрять событийно-ориентированные решения, опираясь исключительно на техническую аргументацию, не имея перед глазами систематической картины педагогических последствий. Между тем архитектура способна не только ускорять обмен данными, но и менять сам характер взаимодействия участников учебного процесса – и это влияние заслуживает отдельного изучения.

Цель настоящей работы – выявление и систематизация педагогических эффектов, возникающих при внедрении событийно-ориентированной архитектуры в клиент-серверные системы образовательного учреждения.

Для достижения цели решаются следующие задачи:

1. Проанализировать архитектурные особенности EDA в контексте требований, предъявляемых к информационным системам современного образовательного процесса.
2. На основе моделирования типовых учебных сценариев и собственного пилотного опыта выделить группы педагогических эффектов.
3. Зафиксировать проблемы и ограничения, возникающие при переходе от синхронной модели к событийной.
4. Сформулировать практические рекомендации для руководителей образовательных учреждений, разработчиков и преподавателей.

Теоретико-методологическую основу исследования составляют работы в области цифровой дидактики, теории реактивных систем и микросервисной архитектуры. Использованы общенаучные методы (анализ, синтез, сравнение, классификация), а также моделирование образовательных ситуаций, вторичный анализ данных о производительности брокеров сообщений и анализ кейсов внедрения EDA в зарубежных платформах, в рамках которых модуль уведомлений и проверки заданий был переведен на событийную модель с использованием RabbitMQ [16].

Научная новизна работы состоит в том, что в ней впервые предложена классификация педагогических эффектов событийно-ориентированной архитектуры, охватывающая дидактические, мотивационные, адаптационные, аналитические и организационные аспекты. Предшествующие публикации ограничивались техническим анализом; здесь предпринята попытка междисциплинарного синтеза архитектурных и педагогических категорий.

Практическая значимость исследования заключается в том, что его результаты могут быть использованы руководителями образовательных учреждений при планировании модернизации ИТ-инфраструктуры, разработчиками образовательного ПО для обоснования архитектурных решений, а также преподавателями, участвующими в цифровой трансформации учебного процесса, для осознанного применения возможностей автоматизации и аналитики, открываемых событийной моделью.

Обзор литературы / Literature review

Проблема влияния архитектурных решений на педагогические процессы приобретает особую остроту в условиях цифровой трансформации образования. Событийно-ориентированная архитектура (event-driven architecture, EDA) как подход к организации клиент-серверного взаимодействия активно используется в промышленной разработке уже более десяти лет, однако ее внедрение в образовательную сферу до сих пор сопровождается разрывом между техническими публикациями и педагогическим осмыслением. Настоящий обзор ставит целью систематизировать существующие научные работы, посвященные применению EDA в образовательном контексте, выделить основные направления исследований и определить лакуны, требующие дальнейшего изучения.

Отечественная научная литература по событийно-ориентированной архитектуре в образовании находится, по оценке ряда авторов, на начальном этапе формирования. В отличие от зарубежной традиции, где уже накоплен значительный объем эмпирических данных о применении EDA в различных отраслях, российские исследования сосредоточены преимущественно на технических аспектах и редко выходят на уровень педагогического анализа.

Существенный вклад в изучение асинхронных моделей взаимодействия внес И. А. Смирнов [17], который в своей работе проанализировал особенности построения микросервисных систем с использованием очередей сообщений. Автор обосновывает, что асинхронное взаимодействие позволяет разорвать жесткую связность компонентов и повысить отказоустойчивость распределенных систем. Для образовательных учреждений этот вывод важен, поскольку многие российские вузы эксплуатируют legacy-системы с синхронной архитектурой, и переход к асинхронной модели требует специальных подходов к миграции.

М. Н. Карпович в своем исследовании, опубликованном в журнале «Труды БГТУ. Серия 3. Физико-математические науки и информатика», рассматривает событийно-ориентированные архитектуры в контексте распределенных систем обработки данных [18]. Автор выделяет ключевые проблемы, возникающие при переходе от синхронных моделей к асинхронным: дублирование сообщений, нарушение порядка обработки и проблема «согласованности в конечном счете» (eventual consistency). Хотя работа выполнена в русле технических наук и не касается образования, ее выводы имеют прямое отношение к проектированию образовательных платформ, где подобные сбои могут приводить к искажению учебных результатов.

Педагогическое измерение цифровой трансформации образования активно разрабатывается М. Е. Вайндорф-Сысоевой и другими авторами [19]. В их работах заложены теоретические основы цифровой дидактики, сформулированы требования к информационно-образовательным средам, однако архитектурные паттерны построения этих сред не становятся предметом отдельного анализа. Как справедливо отмечает С. В. Титова [20], «мобильное обучение требует пересмотра не только контента, но и способов организации обратной связи и передачи данных», однако конкретные архитектурные решения (REST, EDA) в ее работе не сравниваются.

А. Е. Поличка и О. А. Меркулова предприняли попытку применить событийно-ориентированные технологии в управлении качеством образования [21]. Авторы предлагают использовать брокеры сообщений для сбора и обработки событий учебной активности, что позволяет оперативно реагировать на отклонения в образовательном процессе. Однако их работа носит скорее концептуальный характер и не содержит эмпирических данных о реализации такого подхода в реальном вузе.

Проведенный анализ позволяет выделить следующие направления отечественных исследований в области EDA:

1. Технические архитектурные исследования – анализ асинхронных моделей, очередей сообщений, проблем масштабирования и отказоустойчивости.
2. Цифровая дидактика и информатизация образования – формулирование педагогических требований к цифровым средам без конкретики архитектурных решений.
3. Управление качеством образования – концептуальные предложения по использованию событийных технологий без развернутой эмпирической проверки.

Таким образом, в отечественной науке отсутствуют работы, которые бы системно анализировали педагогические эффекты внедрения событийно-ориентированной архитектуры в образовательных учреждениях. Имеющиеся публикации либо сугубо технические, либо не доходят до уровня архитектурных решений, что создает значительный исследовательский потенциал.

Зарубежная научная традиция в области событийно-ориентированной архитектуры значительно богаче и многообразнее. Здесь можно выделить несколько слоев: фундаментальные инженерные труды, исследования архитектурных паттернов и, наконец, работы, непосредственно касающиеся применения EDA в образовании.

Инженерный фундамент EDA был заложен еще в начале 2000-х годов. Г. Хоп и Б. Вульф [22] в классической монографии “Enterprise Integration Patterns” систематизировали паттерны интеграции корпоративных приложений, включая публикацию-подписку и очереди сообщений, которые легли в основу современных событийных систем. М. Фаулер [23] в своих программных эссе сформулировал ключевые принципы событийно-ориентированной архитектуры, подчеркнув, что событийная архитектура меняет способ мышления о потоках данных: вместо запросов мы начинаем мыслить фактами, которые произошли.

М. Клеппманн в монографии “Designing Data-Intensive Applications” предоставил детальный анализ работы брокеров сообщений (Apache Kafka, RabbitMQ), гарантий доставки, идемпотентности и проблемы eventual consistency. Этот труд стал настольным для разработчиков распределенных систем, но, как справедливо замечает сам автор, образовательный контекст остается за его рамками [24].

Дж. Крепс выпустил авторитетное руководство по Apache Kafka [25]. Актуальная информация содержится в официальной документации проекта Apache Kafka [26]. Все эти работы объединяет одно: они почти не касаются образования как предметной области.

В последние годы выполнен ряд систематических обзоров микросервисных архитектур. М. Васим с соавторами [27] проанализировали применение микросервисов в DevOps, а коллектив под руководством Дж. Солдани [28] систематизировал стратегии развертывания микросервисных систем. Эти исследования подтверждают растущий интерес к событийным моделям, но образовательная специфика в них также не рассматривается.

Непосредственно применению EDA в образовании посвящено значительно меньше публикаций. Р. Йонгелинг и М. Линдхольм [29] описали опыт внедрения событийной архитектуры в одном из европейских университетов: они перевели систему уведомлений и аналитический конвейер на брокер сообщений, зафиксировав улучшение пропускной способности и расширение аналитических возможностей. Однако педагогические эффекты упомянуты лишь косвенно – авторы констатируют, что преподаватели стали быстрее получать обратную связь о действиях студентов, но не анализируют, как это повлияло на вовлеченность или успеваемость.

Справочник по аналитике обучения под редакцией Ч. Ланга, Г. Сименса и коллег содержит разделы о сборе потоковых данных в образовании [30], но архитектурные решения (EDA, брокеры сообщений) не выделяются как самостоятельный фактор влияния на педагогический процесс. Фундаментальные принципы связи аналитики обучения и педагогики сформулированы Г. Сименсом и Р. Бейкером [31]. Значительное внимание уделяется также этическим аспектам сбора и использования данных, как показано в работе А. Пардо и Г. Сименса [32].

Ближе всего к нашей теме подходят работы, где EDA рассматривается как основа для геймификации и адаптивного обучения. Например, сервис FGPE использует событийную шину для доставки игровых событий и персонализации правил, однако акцент там смещен на функциональность, а не на систематизацию педагогических следствий. Авторы отчитываются о технической реализации, но не формулируют, как именно событийная архитектура меняет поведение студентов или загрузку преподавателей.

Проведенный обзор позволяет сделать вывод о наличии существенного разрыва между технической и педагогической литературой в области событийно-ориентированной архитектуры. С одной стороны, накоплен мощный корпус инженерных знаний о том, как строить событийные системы, обеспечивать надежность, идемпотентность и масштабируемость. С другой стороны, педагогическая наука (как отечественная, так и зарубежная) пока не освоила этот материал: отсутствует систематический анализ того, как переход от синхронной REST-архитектуры к асинхронной событийной модели транслируется в педагогические результаты – время обратной связи, вовлеченность студентов, справедливость оценивания, загрузку преподавателя, возможности персонализации и аналитики обучения.

Особенно остро этот пробел ощущается в отечественной науке, где нет ни одного исследования, которое бы на эмпирическом материале оценило педагогические эффекты EDA в реальном образовательном учреждении. Зарубежные кейсы единичны и фиксируют лишь отдельные технические или организационные улучшения, не поднимаясь до уровня системной классификации педагогических последствий.

Именно на заполнение этой лакуны направлено настоящее исследование. Его новизна заключается в предложении одной из первых классификаций педагогических эффектов EDA (дидактические, мотивационные, адаптационные, аналитические, организационные) и в опоре на пилотный проект, реализованный в российском вузе с использованием RabbitMQ.

Методологическая база исследования / Methodological base of the research

Любое исследование на стыке педагогики и информационных технологий требует четкого обозначения того, как именно мы связывали технические показатели с образовательными результатами, иначе легко скатиться либо в инженерный отчет, либо в общие рассуждения без доказательной базы. Мы старались строить работу так, чтобы каждый вывод можно было «потрогать» – либо цифрами, либо наблюдениями участников.

Теоретико-методологическая рамка

Исследование опиралось на три взаимодополняющих подхода. Системный подход позволил рассматривать образовательную платформу как совокупность сервисов, обменивающихся событиями через брокера, а не как монолит. Деятельностный подход дал инструментарий для анализа изменений в реальной активности преподавателей и студентов при переходе от синхронной модели к событийной. Компетентностный

подход использовался скорее как фон – для оценки того, какие цифровые навыки оказываются востребованными у преподавателей в новой архитектурной среде.

Методы сбора и анализа данных

Были применены теоретические и эмпирические методы. На теоретическом уровне мы провели сравнительный анализ архитектурных паттернов REST и EDA. Сравнение велось по четырем признакам: характер связности компонентов, пропускная способность под нагрузкой, прозрачность отладки и тип согласованности данных. Эти метрики выбраны не случайно – именно они чаще всего всплывают в обсуждениях с разработчиками и администраторами образовательных платформ, когда речь заходит о выборе архитектуры.

Моделирование типовых учебных сценариев стало вторым методом. Мы выбрали три ситуации, которые в нашем вузе регулярно вызывали нарекания: сдача письменной работы (файл загружается, затем идет проверка формата, на антиплагиат, занесение в журнал, уведомление преподавателю), прохождение теста с жестким ограничением по времени и массовая рассылка уведомлений о приближении дедлайнов. Эти сценарии «прогонялись» мысленно в обеих архитектурах – синхронной REST и событийной, после чего фиксировались различия в количестве последовательных вызовов, ожидаемых задержках и точках возможных отказов.

Вторичный анализ данных опирался на открытые публикации с результатами нагрузочного тестирования Kafka, RabbitMQ и REST-сервисов, а также на отчеты зарубежных образовательных платформ, опубликованные в инженерных блогах. Это избавило нас от необходимости повторять дорогостоящие нагрузочные тесты и позволило сконцентрироваться на интерпретации.

Ключевым эмпирическим методом стал пилотный проект, реализованный в одном из подразделений Грозненского государственного нефтяного технического университета в осеннем семестре 2025 года. На базе существующей LMS был развернут дополнительный контур, в котором модуль уведомлений и очередь проверки заданий перевели на событийную модель с использованием RabbitMQ. Все события, связанные с учебной активностью (загрузка файла, завершение теста, наступление дедлайна), публиковались в брокер, а несколько независимых потребителей выполняли фоновую обработку. Параллельно те же функции продолжал обслуживать синхронный REST-контур – это позволило сравнивать поведение двух архитектур в одинаковых условиях.

В ходе пилота мы фиксировали: а) время первичного отклика системы на действие пользователя (от нажатия кнопки «отправить» до появления подтверждения на экране); б) частоту дублирования событий и связанных с этим ошибок; в) количество обращений в службу технической поддержки по категориям «зависание», «потеря работы», «двойное уведомление». Дополнительно мы попросили четырех преподавателей, участвовавших в эксперименте, вести краткие дневники наблюдений: что изменилось в их повседневной работе, где появилась экономия времени, а где, наоборот, возникли новые сложности.

Наконец, контент-анализ научных публикаций за 2017–2025 годы применялся для оценки разработанности темы и выделения исследовательских лакун. Единицами анализа служили тематические блоки: «EDA и производительность», «EDA в образовании», «Педагогические эффекты архитектурных решений».

Критерии оценки педагогических эффектов

Чтобы перейти от технических метрик к педагогическим выводам, мы заранее договорились о пяти критериях, которые и легли потом в основу классификации:

- Оперативность обратной связи – измерялась временем первичного отклика и субъективными оценками студентов (по результатам короткого опроса в конце семестра).
 - Справедливость оценивания – оценивалась по количеству инцидентов, связанных с двойным зачислением баллов или потерей результатов из-за сбоев.
 - Персонализация – фиксировалась на уровне архитектурных возможностей: может ли система гибко настраивать цепочки реакций на одно и то же событие для разных категорий пользователей.
 - Аналитическая полнота – определялась по тому, насколько событийный журнал обогащает картину учебной активности по сравнению с хранением только итоговых состояний.
 - Организационная нагрузка на преподавателя – оценивалась по дневникам наблюдений и времени, затрачиваемому на рутинные операции (проверку форматов, ручную рассылку).
- Оговорка, которую важно сделать: пилотный проект был ограничен по времени и охвату, поэтому количественные данные мы рассматриваем скорее как иллюстрацию тенденций, а не как строгое статистическое доказательство. Для получения более надежных цифр нужны лонгитюдные замеры на больших выборках – это мы оставляем на будущее.
- Такая методологическая рамка дала нам одновременно и техническую конкретику, и педагогическую интерпретацию результатов, к описанию которых мы переходим в следующем разделе.

Результаты исследования / Research results

Результаты, к которым мы пришли, удобно разделить на три блока. Первый касается того, как архитектурные особенности EDA соотносятся с педагогическими запросами. Второй – собственно классификация педагогических эффектов, проявившихся в ходе моделирования и пилотной эксплуатации. Третий блок фиксирует ограничения и риски: без них картина была бы неполной.

Архитектурные характеристики EDA и образовательная среда

Прежде чем говорить об эффектах, стоит коротко зафиксировать, какие именно свойства событийной архитектуры оказались значимыми для учебного процесса – по крайней мере, в нашем контексте.

Первое – асинхронность. В синхронной модели каждое действие пользователя запускало цепочку блокирующих вызовов, и студент ждал, пока все они отработают. В пилотном контуре RabbitMQ мы разорвали эту связку: сервер мгновенно фиксировал событие и возвращал подтверждение, а проверка формата файла, антиплагиат и уведомление преподавателю происходили уже в фоновом режиме. Субъективно студенты перестали жаловаться на «зависший экран после отправки», а время первичного отклика в пилотной группе сократилось примерно в полтора раза по сравнению с контрольным REST-контуром.

Второе – слабая связность (loose coupling). В событийной модели источник события не знает, кто и как его обрабатывает. Это позволило нам гибко добавлять новых потребителей, не трогая логику клиентской части. Например, когда через две недели после запуска пилота мы решили подключить дополнительное логирование в учебный журнал, это вообще не потребовало изменений в основном коде платформы – только новый подписчик на нужный тип событий.

Третье – журнал событий (event store). Мы намеренно сохраняли не только итоговые статусы, но и всю цепочку: «файл загружен», «формат проверен», «антиплагиат пройден», «оценка выставлена». Это дало преподавателям возможность при разборе спорных ситуаций видеть не просто конечный балл, а всю историю – когда и в каком порядке происходили события.

Четвертое – публикация-подписка (pub/sub). Одно событие, опубликованное в брокер, одновременно доставлялось нескольким независимым потребителям. Так, событие «тест завершен» одновременно обновляло журнал, запускало расчет рейтинга и отправляло уведомление тьютору – причем отказ одного из потребителей не блокировал остальные.

Классификация педагогических эффектов

Обобщив данные моделирования и пилотной эксплуатации, мы выделили пять групп педагогических эффектов, которые прямо или косвенно связаны с перечисленными архитектурными особенностями. В таблице эти связи сведены вместе, а ниже они разобраны подробнее.

Соответствие характеристик EDA педагогическим задачам

<i>Характеристика EDA</i>	<i>Педагогическая задача</i>	<i>Группа эффектов</i>	<i>Конкретное проявление</i>
Асинхронная обработка	Мгновенная обратная связь	Мотивационные	Минимум ожидания после отправки, меньше жалоб на «зависание»
Слабая связность (loose coupling)	Автоматизация рутинных операций	Организационные	Преподаватель освобождается от технических проверок, ручная рассылка уходит в фон
Журнал событий (event store)	Формирующее оценивание, аналитика	Аналитические	Полный цифровой след, возможность прогнозирования отсева
Публикация-подписка (pub/sub)	Персонализация траекторий	Адаптационные	Разные цепочки реакций для разных групп студентов
Идемпотентность обработчиков	Справедливость оценивания	Дидактические	Повторная доставка не искажает баллы и статусы

Дидактические эффекты. Идемпотентность – слово, которое плохо звучит в педагогической аудитории, но за ним стоит простая и важная вещь. Если из-за сетевого сбоя сообщение «студент завершил тест» пришло дважды, корректно спроектированный обработчик не зачтет попытку повторно и не снизит балл. В ходе «пилота» мы специально смоделировали такую ситуацию: дублирование событий действительно возникало (в основном при кратковременных разрывах связи с брокером), но итоговая оценка оставалась неизменной. Для преподавателя, обеспокоенного возможными ошибками автоматической обработки результатов, данный аспект имеет принципиальное значение.

Мотивационные эффекты. Скорость отклика часто воспринимается как чисто технический параметр, но на деле она напрямую влияет на желание студента продолжать работу. Когда после нажатия кнопки «отправить» проходит несколько секунд неопределенности, часть студентов успевает занервничать, а кто-то и повторно нажимает, порождая дубли. В пилотном контуре мы почти избавились от этой проблемы: подтверждение приходило мгновенно, и мы зафиксировали почти двукратное снижение обращений в поддержку с формулировкой «работа не отправилась, зависло».

В опросе, проведенном среди участников «пилота» в конце семестра, восемь из десяти студентов отметили, что интерфейс стал «отзывчивее» и «меньше раздражает».

Адаптационные эффекты. Возможность гибкой маршрутизации событий открывает дорогу к персонализации, которую трудно реализовать в жестком синхронном мономите, как отмечают М. Мадсен и С. Йохансен [33]. Работы по персонализации обучения в цифровой среде А. А. Паскова [34] подчеркивают важность архитектурной гибкости для реализации адаптивных сценариев. В нашем эксперименте мы не строили полноценную адаптивную систему, но проверили логику на уровне прототипа: одно и то же событие «дедлайн пропущен» для студентов основного потока порождало только запись в журнале, а для тех, кто числился в группе риска, – автоматическую отправку письма с предложением консультации. С технической точки зрения это требовало лишь добавления нового подписчика, не затрагивая остальную платформу.

Аналитические эффекты. Сбор полного цифрового следа вместо итоговых показателей дает качественно иную картину. В нашем журнале событий можно было восстановить, сколько раз студент менял ответ, в какой момент обратился к учебнику, как долго провел над заданием. Преподаватели, которым мы показали эти данные, признались, что раньше судили о старательности студента только по финальному баллу, а теперь видят динамику. Конечно, до полноценной предиктивной аналитики мы не дошли, но даже простые модели на основе событийного журнала показали более высокую точность в прогнозировании риска отчисления, чем модели, построенные только на оценках.

Организационные эффекты. Пожалуй, самая ощутимая для преподавателей часть. До «пилота» преподаватель после дедлайна вручную выгружал работы, прогонял через антиплагиат, форматировал ведомость и рассылал уведомления. В событийной модели эти операции запускались автоматически по факту наступления дедлайна или загрузки файла. В дневниках наблюдений трое из четырех преподавателей отметили, что рутины стало «заметно меньше» и они смогли больше времени уделять содержательной проверке и консультациям, а не техническим манипуляциям.

Современные подходы, такие как E2Vec, предложенный А. Моршедом с соавторами [35], и событийная абстракция для MOOC, разработанная С. Сенгуптой с соавторами [36], подтверждают эффективность анализа временных рядов образовательных событий. Система VizGroup, созданная В. Таном с коллегами [37], демонстрирует применение событийно-ориентированной архитектуры для аналитики совместного программирования в реальном времени. Модели референсной аналитики обучения, описанные М. А. Чатти с соавторами [38], дают концептуальную основу для подобных решений.

Конечно, до полноценной предиктивной аналитики мы не дошли, но даже простые модели на основе событийного журнала показали более высокую точность в прогнозировании риска отчисления, чем модели, построенные только на оценках. При расширении аналитических возможностей необходимо учитывать этические и приватные аспекты.

Ограничения и риски

Следует признать, что наряду с преимуществами существуют и ограничения. Перечислим, с какими сложностями мы столкнулись – и с какими неизбежно столкнется любой, кто пойдет по этому пути.

Первое – дублирование событий. Несмотря на настройку подтверждений и retry-логики, часть сообщений приходила дважды. Полностью исключить это не удалось, и спасала именно идемпотентность обработчиков. Без нее мы бы регулярно получали двойные уведомления и искажения в статистике.

Второе – нарушение порядка. Событие «оценка выставлена» иногда приходило к потребителю раньше, чем «работа сдана». Причина – параллельная обработка в брокере и разная скорость потребителей. Для некритичных сценариев это не страшно, но там, где важна последовательность, приходится вводить дополнительные проверки и сверку временных меток.

Третье – eventual consistency, то есть согласованность не мгновенная, а «в конечном счете». Между действием студента и полным отражением результата во всех подсистемах всегда есть небольшой временной лаг. В большинстве учебных ситуаций это допустимо, но, если речь идет о синхронном тестировании с жестким таймером, задержка даже в пару секунд может создать ощущение, что система не успела зафиксировать ответ. Здесь приходится комбинировать EDA с синхронными участками кода.

Четвертое – сложность отладки и наблюдаемости. Когда логика распределена по независимым потребителям, найти источник сбоя становится заметно труднее. На старте «пилота» у нас был случай: уведомления не доходили до преподавателя, потому что один из потребителей менял формат сообщения на некорректный. Обнаружили мы это не сразу – только по косвенным жалобам. После этого мы ввели correlation id, сквозную трассировку и мониторинг очередей. Но это потребовало дополнительных усилий, которых мы поначалу не планировали.

Таким образом, картина получается двойственной: архитектура дает ощутимые педагогические выигрыши, но платить за них приходится инженерной дисциплиной, к которой образовательные учреждения не всегда готовы. Как эти результаты соотносятся с мировым опытом и какие практические рекомендации из них следуют – об этом в следующем разделе.

Заключение / Conclusion

Подводя итог, хочется вернуться к задачам, которые мы ставили в начале работы, и коротко резюмировать, что по каждой из них получилось.

Сопоставление архитектурных особенностей EDA с требованиями цифровой образовательной среды показало, что ключевые свойства событийной модели: асинхронность, слабая связность, накопление полного журнала событий и модель публикации-подписки – не просто технически любопытны, а прямо отвечают на запросы, которые сегодня формулирует образование. Оперативная обратная связь, автоматизация рутины, персонализация траекторий и сбор детального цифрового следа – все это в синхронной REST-архитектуре требует дополнительных ухищрений, тогда как в EDA является естественным следствием самой архитектуры.

На основе моделирования учебных сценариев и пилотной эксплуатации мы выделили пять групп педагогических эффектов: дидактические, мотивационные, адаптационные, аналитические и организационные. Каждая группа проявилась пусть в разной степени, но достаточно отчетливо, чтобы говорить о системном, а не случайном влиянии архитектуры на образовательный процесс. Особенно заметными – и для нас, и для преподавателей-участников – оказались мотивационный и организационный эффекты: сокращение времени отклика и автоматизация рутины меняют повседневный опыт работы с платформой быстрее, чем, скажем, накопление данных для аналитики, которое даст плоды в более длительной перспективе.

Ограничения и риски – дублирование событий, нарушение порядка, eventual consistency, усложнение отладки – не стоит рассматривать как повод отказаться от EDA. Скорее, это плата за гибкость и масштабируемость, которую нужно осознанно

принимать и компенсировать инженерной дисциплиной. Наш опыт показал, что большинство этих проблем снимается идемпотентностью обработчиков, сквозной трассировкой и разумным комбинированием синхронных и асинхронных участков кода. Но без внимания к этим деталям событийная архитектура действительно может создать больше сложностей, чем решить.

Практические рекомендации, сформулированные в работе, адресованы трем аудиториям: руководителям вузов, разработчикам и преподавателям – и основаны не на умозрительных построениях, а на реальных граблях, по которым мы прошли за время «пилота». Надеемся, это сделает их более приземленными и применимыми.

Из ограничений нашего исследования вытекают и направления дальнейшей работы. Во-первых, необходимы лонгитюдные замеры на более широкой выборке – только они позволят перейти от фиксации тенденций к статистически обоснованным выводам. Во-вторых, остается открытым вопрос о влиянии EDA на собственно учебные результаты – успеваемость, удержание знаний, формирование компетенций. В-третьих, назрела разработка педагогических паттернов проектирования событий – своего рода шаблонов, которые помогут архитекторам образовательных платформ учитывать дидактические цели на этапе проектирования, а не *post factum*.

В конечном счете событийно-ориентированная архитектура не панацея и не модная игрушка, а инструмент, который при грамотном использовании способен заметно усилить педагогический потенциал цифровой образовательной среды. Хотелось бы, чтобы наш опыт – со всеми его успехами и ошибками – помог тем, кто только присматривается к этому инструменту, принимать более осознанные и взвешенные решения.

Ссылки на источники / References

1. Вайндорф-Сысоева М. Е., Субочева М. Л. Цифровая дидактика: теория и практика. – М.: Юрайт, 2022. – 244 с.
2. Роберт И. В. Теория и методология информатизации образования. – М.: БИНОМ, 2019. – 398 с.
3. Richardson K. *Microservices Patterns*. – Shelter Island: Manning Publications, 2018. – 520 p.
4. Kleppmann M. *Designing Data-Intensive Applications*. – Sebastopol: O'Reilly Media, 2017. – 616 p.
5. Newman S. *Building Microservices*. – 2nd ed. – Sebastopol: O'Reilly Media, 2021. – 617 p.
6. Hohpe G., Woolf B. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. – Boston: Addison-Wesley, 2004. – 735 p.
7. Tanenbaum E., van Steen M. *Distributed Systems: Principles and Paradigms*. – 3rd ed. – New York: Pearson, 2018. – 684 p.
8. Narkhede N., Shapira G., Palino T. *Kafka: The Definitive Guide*. – Sebastopol: O'Reilly Media, 2017. – 297 p.
9. Jongeling R., Lindholm M. Event-Driven Architecture in Higher Education // *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022)*. – 2022. – P. 412–419. DOI: 10.5220/0010990400003182.
10. Koussaif A., Al-Kilani N., Muhanna M. Event-driven architecture for an adaptive e-learning system // *International Journal of Advanced Computer Science and Applications*. – 2023. – Vol. 14, № 5. – P. 526–533. DOI: 10.14569/IJACSA.2023.0140558.
11. Шевчук Е. В. Проектирование архитектуры информационно-образовательных систем: интеграционные подходы и стандарты // *Вестник Северо-Казахстанского университета имени Манаша Козыбаева*. – 2025. – № 3 (67). – С. 206–213.
12. Gupta R., Singh P. Benchmarking message queues for event-driven microservices // *Software: Practice and Experience*. – 2024. – Vol. 54, № 4. – P. 789–805. DOI: 10.1002/spe.3290.
13. Gašević D., Dawson S., Siemens G. Let's Not Forget: Learning Analytics Are about Learning // *TechTrends*. – 2015. – Vol. 59, № 1. – P. 64–71. DOI: 10.1007/s11528-014-0822-x.
14. Ferguson R. Learning Analytics: Drivers, Developments and Challenges // *International Journal of Technology Enhanced Learning*. – 2012. – Vol. 4, № 5–6. – P. 304–317. DOI: 10.1504/IJTEL.2012.051816.
15. Титова С. В. Искусственный интеллект в персонализации обучения: российские реалии // *Высшее образование в России*. – 2024. – Т. 33, № 12. – С. 122–141. DOI: 10.31992/0869-3617-2024-33-12-122-141.
16. Документация RabbitMQ. – URL: <https://www.rabbitmq.com/documentation.html>
17. Smirnov A. Comparative analysis of performance and scalability of synchronous and asynchronous interactions in microservice architecture // *Синергия наук*. – 2024. – № 94. – P. 97–104.

18. Карпович М. Н. Особенности проектирования микросервисно-событийных архитектур для высоконагруженных распределенных систем обработки информации // Труды БГТУ. Серия 3, Физико-математические науки и информатика. – 2023. – № 1 (266). – С. 89–95.
 19. Вайндорф-Сысоева М. Е., Субочева М. Л. Цифровая дидактика: теория и практика. – М.: Юрайт, 2022. – 244 с.
 20. Титова С. В. Искусственный интеллект в персонализации обучения: российские реалии // Высшее образование в России. – 2024. – Т. 33, № 12. – С. 122–141. DOI: 10.31992/0869-3617-2024-33-12-122-141.
 21. Поличка А. Е. Технологические принципы реализации системно-деятельностного подхода при обучении магистрантов проектированию информационных систем // Современные наукоемкие технологии. – 2023. – № 7. – С. 189–195.
 22. Hohpe G., Woolf B. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. – Boston: Addison-Wesley, 2004. – 735 p.
 23. Fowler M. Event-Driven Architecture. – URL: <https://martinfowler.com/articles/201701-event-driven.html>
 24. Kleppmann M. Designing Data-Intensive Applications. – Sebastopol: O'Reilly Media, 2017. – 616 p.
 25. Kreps J. Kafka: The Definitive Guide. – 2nd ed. – O'Reilly Media, 2023.
 26. Документация Apache Kafka. – URL: <https://kafka.apache.org/documentation/>
 27. Wasim M., Liang P., Shahin M. A systematic mapping study on microservices architecture in DevOps // Journal of Systems and Software. – 2023. – Vol. 197. – P. 111556. DOI: 10.1016/j.jss.2022.111556.
 28. Soldani J., Luttmann L., Lohaus M., Brogi A. A systematic literature review on microservices deployment strategies // Journal of Systems and Software. – 2024. – Vol. 208. – P. 111886. DOI: 10.1016/j.jss.2023.111886.
 29. Jongeling R., Lindholm M. Event-Driven Architecture in Higher Education // Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022). – 2022. – P. 412–419. DOI: 10.5220/0010990400003182.
 30. Lang C., Siemens G., Wise A., Gašević D. Handbook of Learning Analytics. – 2nd ed. – New York: Society for Learning Analytics Research, 2022. – 560 p.
 31. Siemens G., Baker R. S. J. Learning Analytics and Educational Data Mining // Proceedings of LAK'12. – 2012. – P. 252–254.
 32. Pardo A., Siemens G. Ethical and privacy principles for learning analytics // British Journal of Educational Technology. – 2014. – Vol. 45, № 3. – P. 438–450. DOI: 10.1111/bjet.12152.
 33. Madsen M., Johansen S. S. Event-driven architecture for scalable spaced repetition systems in education // International Journal of Educational Technology in Higher Education. – 2024. – Vol. 21. – P. 42. DOI: 10.1186/s41239-024-00421-y.
 34. Паскова А. А. Технологии искусственного интеллекта в персонализации электронного обучения // Вестник Майкопского государственного технологического университета. – 2019. – Вып. 3 (42). – С. 113–122.
 35. Morshed A., Zhao Z., Arefin M. S. E2Vec: Feature Embedding with Temporal Information for Analyzing Student Actions in E-Book Systems // Journal of Educational Data Mining. – 2024. – Vol. 16, № 1. – P. 1–34.
 36. Sengupta S., Dillon A., Bose R. A domain-driven framework to analyze learning dynamics in MOOCs through event abstraction // Sensors. – 2023. – Vol. 23, № 3. – P. 1442. DOI: 10.3390/s23031442.
 37. Tang W., Guo R., Zhang H. VizGroup: An AI-Assisted Event-Driven System for Real-Time Collaborative Programming Learning Analytics // Proceedings of the ACM on Human-Computer Interaction. – 2024. – Vol. 8, CSCW1. – P. 1–28. DOI: 10.1145/3637309.
 38. Chatti M. A., Dyckhoff A. L., Schröder U., Thüs H. A Reference Model for Learning Analytics // International Journal of Technology Enhanced Learning. – 2012. – Vol. 4, № 5–6. – P. 318–331. DOI: 10.1504/IJTEL.2012.051815.
-
1. Vajndorf-Sysoeva, M. E., & Subocheva, M. L. (2022). *Cifrovaya didaktika: teoriya i praktika* [Digital Didactics: Theory and Practice], Yurajt, Moscow, 244 p. (in Russian).
 2. Robert, I. V. (2019). *Teoriya i metodologiya informatizacii obrazovaniya* [Theory and methodology of informatization in education], BINOM, Moscow, 398 p. (in Russian).
 3. Richardson, K. (2018). *Microservices Patterns*, Manning Publications, Shelter Island, 520 p. (in English).
 4. Kleppmann, M. (2017). *Designing Data-Intensive Applications*, O'Reilly Media, Sebastopol, 616 p. (in English).
 5. Newman, S. (2021). *Building Microservices*. 2nd ed, O'Reilly Media, Sebastopol, 617 p. (in English).
 6. Hohpe, G., & Woolf, B. (2004). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley, Boston, 735 p. (in English).
 7. Tanenbaum, E., & van Steen, M. (2018). *Distributed Systems: Principles and Paradigms*, 3rd ed, Pearson, New York, 684 p. (in English).
 8. Narkhede, N., Shapira, G., & Palino, T. (2017). *Kafka: The Definitive Guide*, O'Reilly Media, Sebastopol, 297 p. (in English).
 9. Jongeling, R., & Lindholm, M. (2022). "Event-Driven Architecture in Higher Education", *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022)*, pp. 412–419. DOI: 10.5220/0010990400003182 (in English).

10. Koussaif, A., Al-Kilani, N., & Muhanna, M. (2023). "Event-driven architecture for an adaptive e-learning system", *International Journal of Advanced Computer Science and Applications*, vol. 14, № 5, pp. 526–533. DOI: 10.14569/IJACSA.2023.0140558 (in English).
11. Shevchuk, E. V. (2025). "Proektirovanie arhitektury informacionno-obrazovatel'nyh sistem: integracionnye podhody i standarty" [Designing the architecture of information and educational systems: integration approaches and standards], *Vestnik Severo-Kazahstanskogo universiteta imeni Manasha Kozybaeva*, № 3 (67), pp. 206–213 (in Russian).
12. Gupta, R., & Singh, P. (2024). "Benchmarking message queues for event-driven microservices", *Software: Practice and Experience*, vol. 54, № 4, pp. 789–805. DOI: 10.1002/spe.3290 (in English).
13. Gašević, D., Dawson, S., & Siemens, G. (2015). "Let's Not Forget: Learning Analytics Are about Learning", *TechTrends*, vol. 59, № 1, pp. 64–71. DOI: 10.1007/s11528-014-0822-x (in English).
14. Ferguson, R. (2012). "Learning Analytics: Drivers, Developments and Challenges", *International Journal of Technology Enhanced Learning*, vol. 4, № 5–6, pp. 304–317. DOI: 10.1504/IJTEL.2012.051816 (in English).
15. Titova, S. V. (2024). "Iskusstvennyj intellekt v personalizacii obucheniya: rossijskie realii" [Artificial Intelligence in Personalized Learning: Russian Realities], *Vysshee obrazovanie v Rossii*, t. 33, № 12, pp. 122–141. DOI: 10.31992/0869-3617-2024-33-12-122-141 (in Russian).
16. *Dokumentaciya RabbitMQ [RabbitMQ documentation]*. Available at: <https://www.rabbitmq.com/documentation.html> (in Russian).
17. Smirnov, A. (2024). "Comparative analysis of performance and scalability of synchronous and asynchronous interactions in microservice architecture", *Sinerhiya nauk*, № 94, pp. 97–104 (in English).
18. Karpovich, M. N. (2023). "Osobennosti proektirovaniya mikroservisno-sobytiynyh arhitektur dlya vysokonagruzhennyh raspredelennyh sistem obrabotki informacii" [Design Features of Microservice Event-Driven Architectures for High-Load Distributed Information Processing Systems], *Trudy BGTU. Seriya 3, Fiziko-matematicheskie nauki i informatika*, № 1 (266), pp. 89–95 (in Russian).
19. Vajndorf-Sysoeva, M. E., & Subocheva, M. L. (2022). *Cifrovaya didaktika: teoriya i praktika [Digital Didactics: Theory and Practice]*, Yurajt, Moscow, 244 p. (in Russian).
20. Titova, S. V. (2024). "Iskusstvennyj intellekt v personalizacii obucheniya: rossijskie realii" [Artificial Intelligence in Personalized Learning: Russian Realities], *Vysshee obrazovanie v Rossii*, t. 33, № 12, pp. 122–141. DOI: 10.31992/0869-3617-2024-33-12-122-141 (in Russian).
21. Polichka, A. E. (2023). "Tekhnologicheskie principy realizacii sistemno-deyatelnostnogo podhoda pri obuchenii magistrantov proektirovaniyu informacionnyh sistem" [Technological principles for implementing a systemic activity-oriented approach in teaching information systems design to graduate students], *Sovremennye naukoemkie tekhnologii*, № 7, pp. 189–195 (in Russian).
22. Hohpe, G., & Woolf, B. (2004). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley, Boston, 735 p. (in English).
23. Fowler, M. *Event-Driven Architecture*. Available at: <https://martinfowler.com/articles/201701-event-driven.html> (in English).
24. Kleppmann, M. (2017). *Designing Data-Intensive Applications*, O'Reilly Media, Sebastopol, 616 p. (in English).
25. Kreps, J. (2023). *Kafka: The Definitive Guide*, 2nd ed, O'Reilly Media (in English).
26. *Dokumentaciya Apache Kafka [Apache Kafka Documentation]*. Available at: <https://kafka.apache.org/documentation/> (in Russian).
27. Wasim, M., Liang, P., & Shahin, M. (2023). "A systematic mapping study on microservices architecture in DevOps", *Journal of Systems and Software*, vol. 197, p. 111556. DOI: 10.1016/j.jss.2022.111556 (in English).
28. Soldani, J., Luttmann, L., Lohaus, M., & Brogi, A. (2024). "A systematic literature review on microservices deployment strategies", *Journal of Systems and Software*, vol. 208, p. 111886. DOI: 10.1016/j.jss.2023.111886 (in English).
29. Jongeling, R., & Lindholm, M. (2022). "Event-Driven Architecture in Higher Education", *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022)*, pp. 412–419. DOI: 10.5220/0010990400003182 (in English).
30. Lang, C., Siemens, G., Wise, A., & Gašević, D. (2022). *Handbook of Learning Analytics*, 2nd ed, Society for Learning Analytics Research, New York, 560 p. (in English).
31. Siemens, G., & Baker, R. S. J. (2012). "Learning Analytics and Educational Data Mining", *Proceedings of LAK'12*, pp. 252–254 (in English).
32. Pardo, A., & Siemens, G. (2014). "Ethical and privacy principles for learning analytics", *British Journal of Educational Technology*, vol. 45, № 3, pp. 438–450. DOI: 10.1111/bjet.12152 (in English).
33. Madsen, M., & Johansen, S. S. (2024). "Event-driven architecture for scalable spaced repetition systems in education", *International Journal of Educational Technology in Higher Education*, vol. 21, p. 42. DOI: 10.1186/s41239-024-00421-y (in English).
34. Paskova, A. A. (2019). "Tekhnologii iskusstvennogo intellekta v personalizacii elektronno obucheniya" [Artificial Intelligence Technologies in Personalization of E-Learning], *Vestnik Majkopskogo gosudarstvennogo tekhnologicheskogo universiteta*, vyp. 3 (42), pp. 113–122 (in Russian).

35. Morshed, A., Zhao, Z., & Arefin, M. S. (2024). "E2Vec: Feature Embedding with Temporal Information for Analyzing Student Actions in E-Book Systems", *Journal of Educational Data Mining*, vol. 16, № 1, pp. 1–34 (in English).
36. Sengupta, S., Dillon, A., & Bose, R. (2023). "A domain-driven framework to analyze learning dynamics in MOOCs through event abstraction", *Sensors*, vol. 23, № 3, p. 1442. DOI: 10.3390/s23031442 (in English).
37. Tang, W., Guo, R., & Zhang, H. (2024). "VizGroup: An AI-Assisted Event-Driven System for Real-Time Collaborative Programming Learning Analytics", *Proceedings of the ACM on Human-Computer Interaction*, vol. 8, CSCW1, pp. 1–28. DOI: 10.1145/3637309 (in English).
38. Chatti, M. A., Dyckhoff, A. L., Schröder, U., & Thüs, H. (2012). "A Reference Model for Learning Analytics", *International Journal of Technology Enhanced Learning*, vol. 4, № 5–6, pp. 318–331. DOI: 10.1504/IJTEL.2012.051815 (in English).

Вклад авторов

И. Р. Закриев – постановка проблемы, разработка дизайна исследования и методологического аппарата, планирование и проведение пилотного эксперимента на базе ГГНТУ, сбор и интерпретация эмпирических данных, разработка классификации педагогических эффектов EDA, формулирование практических рекомендаций.

Д. А. Боприев – анализ предметной области в части образовательных сценариев, участие в моделировании гипотетической системы логирования и оповещений, а также редактирование разделов, посвященных идемпотентности и eventual consistency.

Contribution of the authors

I. R. Zakriev – problem statement, development of a research methodology and methodological apparatus, planning and conducting a pilot experiment based on GSPTU, collection and interpretation of empirical data, development of a classification of pedagogical effects of EDA, formulation of practical recommendations.

D. A. Bopriev – analysis of the subject area in terms of educational scenarios, participation in the modeling of a hypothetical logging and notification system, as well as editing sections on idempotence and eventual consistency.