

5.2.6. Менеджмент

Научная статья

УДК 004.413

<https://doi.org/10.37493/2307-907X.2025.3.12>

ОПТИМИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ: РОЛЬ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОСТРОЕНИЯ ИНТЕГРАЦИОННЫХ РЕШЕНИЙ В ПОВЫШЕНИИ ГИБКОСТИ И АДАПТИВНОСТИ ПРЕДПРИЯТИЯ

Анна Юрьевна Орлова^{1*}, Анатолий Александрович Сорокин²

- ¹ Северо-Кавказский федеральный университет (д. 1, ул. Пушкина, Ставрополь, 355017, Российская Федерация)
² Ставропольский государственный аграрный университет (д. 12, пер. Зоотехнический, Ставрополь, 355035, Российская Федерация)
¹ mss.annette@gmail.com; <https://orcid.org/0000-0001-7173-7526>
² sorokin_a_a@bk.ru; <https://orcid.org/0000-0003-4662-7960>
 * Автор, ответственный за переписку

Аннотация. Введение. В условиях современной цифровой трансформации и растущей сложности информационных ландшафтов предприятий, необходимость эффективной интеграции разнородных систем становится критически важной. **Цель.** Представление концепции ИСИР, определение ее ключевых архитектурных принципов, компонентов и возможностей. **Материалы и методы.** Существует множество подходов к интеграции информационных систем, каждый из которых имеет свои преимущества и недостатки: точечная интеграция (Point-to-Point); ESB (Enterprise Service Bus); API Management; iPaaS (Integration Platform as a Service); Microservices Architecture. **Результаты и обсуждение.** Данная статья представляет собой исследование и разработку информационной системы, предназначенной для автоматизации и упрощения процесса построения интеграционных решений с целью создания эффективной среды для работы компании. Система, основанная на модульной архитектуре и использовании метаданных, позволяет визуализировать потоки данных, моделировать интеграционные сценарии и автоматически генерировать интеграционный код. Ключевыми целями разработки являются повышение скорости разработки, снижение затрат и минимизация ошибок, связанных с ручным кодированием. **Заключение.** Результаты тестирования продемонстрировали значительное ускорение процесса интеграции и повышение качества интеграционных решений.

Ключевые слова: интеграция систем, интеграционные решения, информационная система, метаданные, визуальное моделирование, автоматическая генерация кода

Для цитирования: Орлова А. Ю., Сорокин А. А. Оптимизация бизнес-процессов: роль информационной системы построения интеграционных решений в повышении гибкости и адаптивности предприятия // Вестник Северо-Кавказского федерального университета. 2025. № 3 (108). С. 125–132. <https://doi.org/10.37493/2307-907X.2025.3.12>

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Статья поступила в редакцию 05.02.2025;

одобрена после рецензирования 19.03.2025;

принята к публикации 27.03.2025.

Research article

BUSINESS PROCESS OPTIMIZATION: THE ROLE OF THE INFORMATION SYSTEM FOR INTEGRATION SOLUTIONS IN INCREASING THE FLEXIBILITY AND ADAPTABILITY OF THE ENTERPRISE

Anna Yu. Orlova^{1*}, Anatoly A. Sorokin²

¹ North-Caucasus Federal University (1, Pushkin str., Stavropol, 355017, Russian Federation)

² Stavropol State Agrarian University (12, lane Zootechnical, Stavropol, 355035, Russian Federation)

¹ mss.annette@gmail.com; <https://orcid.org/0000-0001-7173-7526>

² sorokin_a_a@bk.ru; <https://orcid.org/0000-0003-4662-7960>

* Corresponding author

Abstract. Introduction. In the context of modern digital transformation and the growing complexity of information landscapes of enterprises, the need for effective integration of heterogeneous systems is becoming critically important. **Goal.** The study aims to present the ISIR concept, identify its key architectural principles, components, and capabilities. **Materials and methods.** There are many approaches to the integration of information systems, each of which has its advantages and disadvantages: Point-to-Point integration; ESB (Enterprise Service Bus); API Management; iPaaS (Integration Platform as a Service); Microservices Architecture. **Results and discussion.** The article is a research and development of an information system designed to automate and simplify the process of building integration solutions in order to create an effective environment for the company. The system, based on a modular architecture and the use of metadata, allows you to visualize data flows, simulate integration scenarios, and automatically generate integration code. The key development goals are to increase the speed of development, reduce costs, and minimize errors associated with manual coding. **Conclusion.** The test results demonstrated a significant acceleration of the integration process and an improvement in the quality of integration solutions.

Keywords: system integration, integration solutions, information system, metadata, visual modeling, automatic code generation

For citation: Orlova AYu, Sorokin AA. Business process optimization: the role of the information system for integration solutions in increasing the flexibility and adaptability of the enterprise. Newsletter of North-Caucasus Federal University. 2025;3(108):125-132. (In Russ.). <https://doi.org/10.37493/2307-907X.2025.3.12>

Conflict of interest: the authors declare no conflicts of interests.

The article was submitted 05.02.2025;

approved after reviewing 19.03.2025;

accepted for publication 27.03.2025.

Введение / Introduction. Интеграция информационных систем является ключевым фактором успеха для современных предприятий. Разрозненные системы, не способные обмениваться данными, приводят к дублированию информации, ошибкам и снижению эффективности бизнес-процессов [1]. Построение эффективных интеграционных решений требует значительных усилий, времени и экспертизы, особенно в условиях гетерогенного ландшафта IT-систем.

Существующие подходы к интеграции часто опираются на ручное кодирование, которое является трудоемким процессом при проектировании и генерировании интеграционных потоков визуально, используя метаданные и модульную архитектуру. Целью разработки является повышение скорости и качества разработки интеграционных решений, а также снижение затрат на их внедрение и поддержку [3].

Предметом исследования является процесс разработки интеграционных решений в рамках построения корпоративных хранилищ данных. Процесс разработки в компании имеет множество подготовленных регламентов, а также несколько разработанных адаптеров. Однако на данный момент они преимущественно относятся к старому технологическому стеку [4].

Как правило, у организаций практически нет выработанного подхода автоматизации построения интеграционных решений в связи с технологическими ограничениями тех средств, которые были актуальны до последних. Из-за этого сам процесс разработки включает в себя много ручного труда.

Кроме того, из-за отсутствия полноценной кодовой базы, которую можно было переиспользовать, на новых проектах могут возникать неожиданные ошибки в работе как технического порядка, так и логического, т. к. даже при единообразии подхода к разработке построения процесса интеграции всё равно занимается разработчик, который способен совершить ошибку при большом объёме одинаковых задач.

Организации необходимо иметь систему, которая будет способна на основании введённых данных об источниках данных автоматически строить загрузку из них в корпоративное хранилище данных, а также проводить аудит загрузки. Это позволит сократить время разработки, сохраняя единые кодовую базу и подход к интеграции.

Материалы и методы исследований / Materials and methods of research. До недавних пор все проекты преимущественно строились на базе технологий Microsoft (MSSQL, SSIS, SSAS, PowerBI), это накладывало определённые ограничения. В частности, потенциал автоматизации крайне ограниченный, а те небольшие имеющиеся возможности делают решение ещё менее гибким и более тяжело поддерживаемым.

Кроме того, в связи с острой необходимостью российского рынка в отказе от западных технологий, распространяемых по лицензии, дальнейшее развитие на данном стеке является нецелесообразным.

На рынке уже представлено несколько решений, например: KORE.DWH, ProPlum, Prophecy. Однако все они имеют серьёзные ограничения. Одним из главных, среди прочих, является и тот факт, что они разработаны компаниями-конкурентами. Таким образом, использование чужих разработок может нести и репутационные издержки для рассматриваемой организации.

Благодаря анализу существующих решений, а также текущего подхода, можно выделить сразу несколько важных технологических аспектов, которые должны учитываться при разработке системы [6]:

- 1) использование технологий с открытым исходным кодом необходимо, чтобы избежать проблемы с лицензированием и завязками на поставщиков;
- 2) возможность интеграции с разнообразными источниками данных, а также поддержка любых целевых систем на основе PL/pgSQL, т. е. и Greenplum, и PostgreSQL;
- 3) наличие сквозного аудита;
- 4) отсутствие фиксации на выбранной модели данных в хранилище;
- 5) возможность встраивания системы в существующие системы заказчика.

Результаты исследований и их обсуждение / Research results and their discussion. Информационная система разрабатывается с целью упрощения интеграции с источниками данных компаний.

Автоматизация процессов внедрения снижает необходимость вручную настраивать и мониторить каждый шаг разработки, что сокращает затраты на труд и снижает риск человеческих ошибок. Кроме того, автоматизация ускоряет внедрение, что позволит компании предоставлять услуги быстрее и эффективнее, благодаря чему будет легче привлечь новых клиентов [5].

Решаемая задача может быть рассмотрена сразу на нескольких уровнях управления в компании. На стратегическом уровне разработка системы важна для сокращения трудозатрат, повышения конкурентоспособности и качества услуг. На тактическом уровне система поможет оптимизировать процессы внедрения и планирования ресурсов компании. На оперативном же уровне обеспечит автоматизацию рутинных задач разработки.

Цель внедрения системы заключается в решении проблем, выявленных ранее, таких как снижение трудозатрат и улучшения качества услуг. Цель можно разбить на две подгруппы [7]:

I. Улучшение экономических показателей:

- 1) уменьшение времени, затрачиваемого на рутинные операции, позволит сократить расходы на оплату труда и повысить его производительность.
- 2) сокращение сроков разработки и стоимости услуг приведёт новых клиентов и увеличит прибыль компании.

II. Улучшение качества обработки информации. Это важная часть цели несмотря на то, что основными её преимуществами будут пользоваться потенциальные заказчики:

1) система сократит время, необходимое для загрузки и обработки данных, что обеспечит оперативную доступность данных;

2) автоматизированная система уменьшит ручное вмешательство и вместе с ним – вероятность ошибок в данных.

Система должна быть способна строить пакеты загрузки данных автоматически на основе введенных метаданных о структурах в источнике. Это позволит исключить из процесса разработки воссоздание алгоритмов загрузки данных (исключается риск некорректной логики), а также освободит разработчика от настройки загрузки из множества однообразных структур данных, будь то таблицы в СУБД, файлы или API [8].

Существует ряд платформ и инструментов, предназначенных для интеграции систем, включая: Enterprise Service Bus (ESB); Integration Platform as a Service (iPaaS); Low-Code/No-Code платформы.

Построение диаграммы требуется для определения основных этапов формирования системы. Это позволит сформировать основные этапы для формирования ИС, определяя необходимые вводные данные и структурные элементы, которые необходимо реализовать в информационной системе. Модель доработанного потока данных представлена на рисунке 1.

На входе система получает метаданные о структурах данных в источнике. Метаданные должны быть получены на основе процесса профилирования данных с целью установления их достоверности и соответствия желаемому результату. Метаданные попадают в специальную базу данных системы.

Впоследствии введенные метаданные сверяются с источником. В случае наличия ошибок (например, указана несуществующая таблица) информация об этом попадет в систему мониторинга и при необходимости вышлет уведомления об ошибке.

На основе метаданных также строятся структуры в хранилище – таблицы с введенным набором атрибутов. После чего автоматизированная загрузка поместит данные в хранилище, используя соответствующие таблицы [2].

Для разработки был выбран стек, состоящий из реляционной PostgreSQL и Airflow, а в качестве архитектуры была выбрана сервис-ориентированная. Сервис-ориентированная архитектура позволит быстро разворачивать решение практически на любых системах при помощи Docker, заменять модули на аналогичные системы заказчика. PostgreSQL выбран ввиду широкого функционала, интеграции со многими сторонними системами, а также совместимости с СУБД Greenplum. Airflow принят как наиболее популярная, гибкая и развитая платформа [10].

База метаданных должна отвечать нескольким задачам:

- 1) хранение метаданных о структурах источников;
- 2) хранение результатов аудита загрузки данных;
- 3) хранение настроенных проверок качества данных.

Отсюда, можно разделить базу на три соответствующие схемы.

Загрузка в Airflow, как и выполнение любых других задач, осуществляется при помощи так называемых операторов. Оператор представляет метод, написанный на Python, который можно переиспользовать из раза в раз. Это позволит единожды подготовить алгоритм загрузки, избавляя разработчиков от необходимости его повторно имплементировать.

Другим важным аспектом работы Airflow является построение DAG'a, ациклического направленного графа. Это позволяет в нужном порядке с нужными условиями запускать необходимые задачи. Данный подход хоть и гибкий, но для неопытного разработчика может оказаться весьма сложным. Однако, Airflow предоставляет возможности динамической генерации задач, чем можно воспользоваться для освобождения разработчика от воспроизведения корректной логики загрузки в хранилище [9].

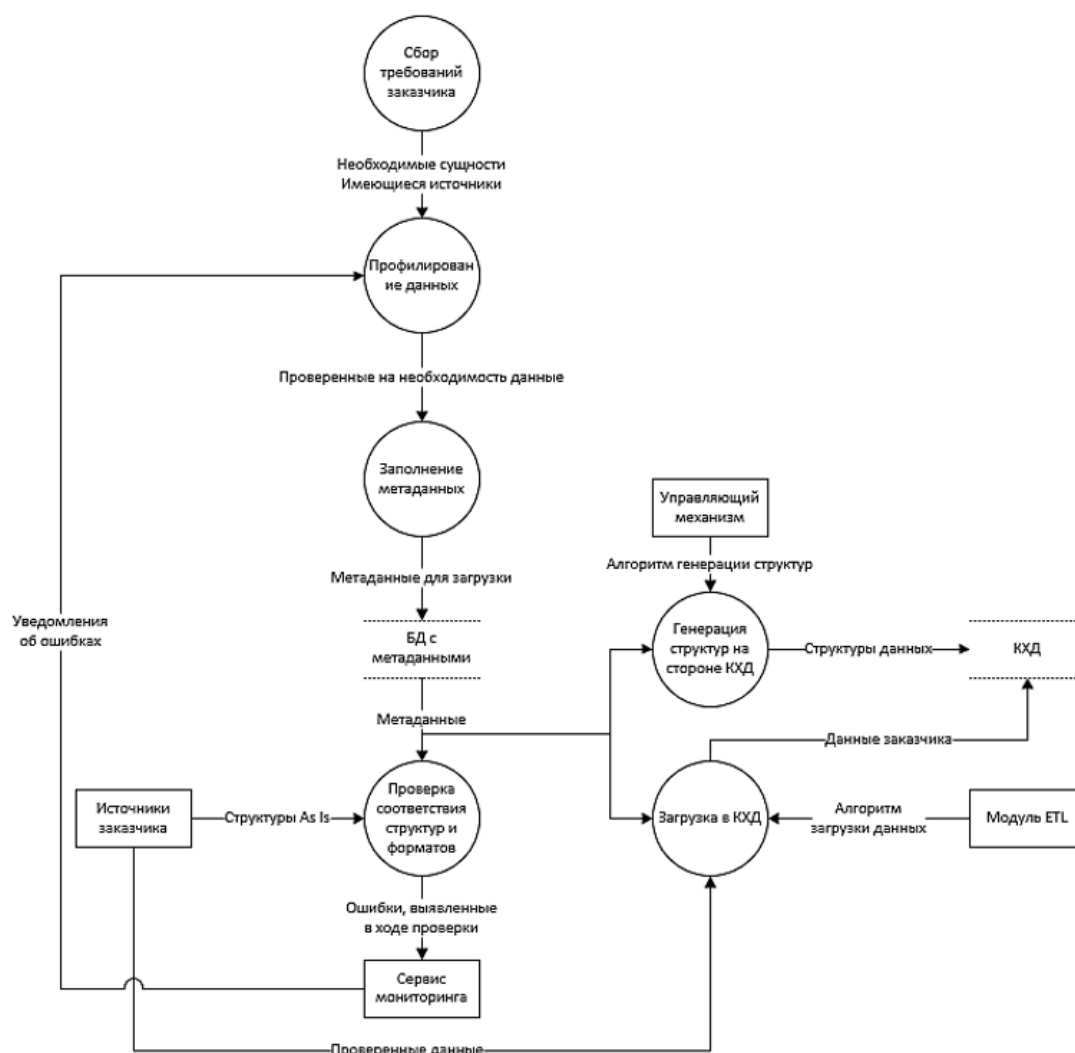


Рис. 1. Диаграмма потоков данных для интеграции с источником / Fig. 1. Data flow diagram for integration with the source

Для динамической генерации будут использованы метаданные, а также заготовленные макросы, которые на основании полученных метаданных, вернут группу задач в нужном порядке и с нужными зависимостями.

Для удобства пользования необходимо предусмотреть как минимум два уровня генерации: на уровне таблицы и на уровне источника. Генерация на уровне источника подойдёт в том случае, если базовый метод загрузки полностью удовлетворяет нуждам интеграции и не нуждается в доработке. Однако возникают ситуации, когда перед загрузкой каждой таблицы нужно выполнить какие-то предварительные операции. Например, скачать файл из необычного источника или подготовить модифицированный запрос к источнику. В таком случае необходим более низовой уровень контроля над загрузкой, что и обеспечит генерация на уровне таблицы.

Были созданы базовые классы, от которых впоследствии будут наследовать классы для конкретных типов источника.

Базовый класс, генерирующий загрузку таблицы, будет включать общие методы, которые обеспечат корректную логику загрузки.

После определения базовых классов можно использовать их для имплементации загрузки из источника.

Можно заметить, что в рамках макроса используется оператор загрузки. Таким образом, пользователю нет необходимости напрямую взаимодействовать с шаблонизированным загрузчиком, требуется лишь корректно заполнить метаданные и вызвать подходящий метод, что серьезно сокращает ручной труд разработчика. Ему остаётся лишь вызвать соответствующий макрос.

В итоге макрос получает лишь название источника, после чего способен сгенерировать всю загрузку на основе метаданных (рисунок 2).

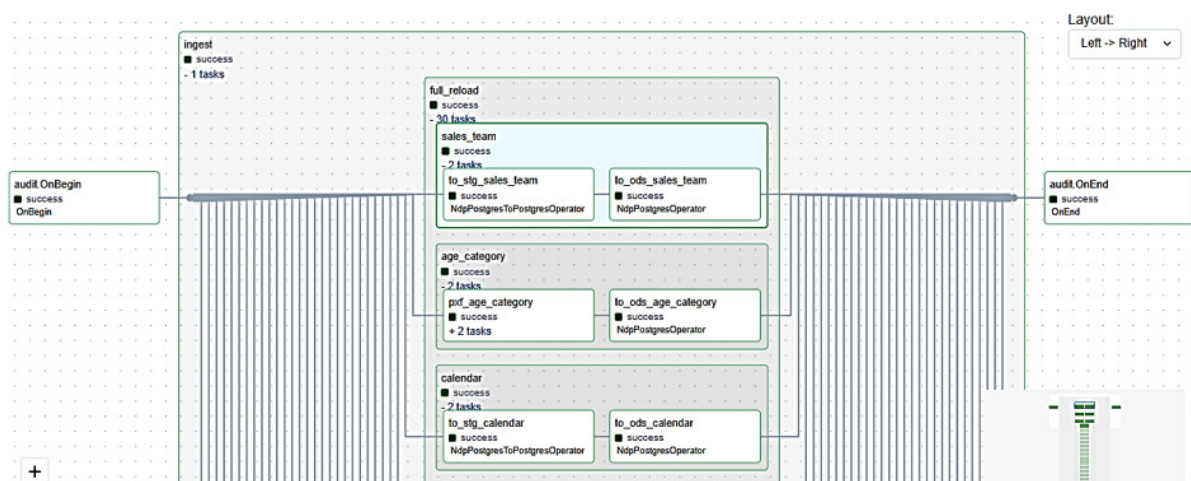


Рис. 2. Сгенерированный граф загрузки из источника в хранилище / Fig. 2. Generated graph of loading from source to storage

Данный модуль позволяет в кратчайшие сроки настроить загрузку из множества таблиц самых разных источников, что позволит значительно сократить трудозатраты и расходы на проекты внедрения.

Другим функциональным блоком является сквозной аудит. Он является важным инструментом для первичного исследования инцидентов, т. к. содержит текст ошибок, а также позволяет однозначно установить, какие данные были загружены в рамках конкретного запуска.

Для того чтобы исключить попадание в базу некорректных данных, был написан блок проверки данных файловых источников, т. к. они часто заполняются либо в ручном формате, либо в системах, которые могут быть слабо связаны с другими системами заказчика, что может приводить к ошибкам в данных.

Результатом работы операторов при наличии ошибок будут соответствующие сообщения в логах (рисунок 3), а также можно настроить почтовую рассылку при наличии SMTP-сервера.

```
INFO - Exporting env vars. AIRFLOW_CTX_DAG_OWNER= AIRFLOW_CTX_DAG_ID=load_excel_to_stg
ing connection ID 'Ecomm' for task execution.
INFO - 2024-05-16 13:17:27.428404|Ecomm.xlsx|Columns {'adi_db'} are missing. Data was not loaded.
INFO - Marking task as SUCCESS. dag_id=load_excel_to_stg, task_id=load_EXCEL_Ecomm.Check_EXCEL_Ec
```

Рис. 3. Результат проверки некорректно заполненного файла / Fig. 3. The result of checking an incorrectly filled file

Заключение / Conclusion. В данной работе была представлена разработка информационной системы для построения интеграционных решений, основанной на визуальном моделировании и автоматической генерации кода. Система позволяет значительно упростить и ускорить процесс интеграции разнородных систем, а также снизить затраты и повысить качество интеграционных

решений. С учетом полученных результатов была разработана специализированная система автоматизации построения интеграций с хранилищами данных. Эта система обеспечивает возможность быстрого и эффективного подключения различных источников данных к аналитическим платформам и системам отчетности.

В качестве дальнейших исследований планируется:

- расширение репозитория метаданных для поддержки большего количества систем и технологий;
- улучшение модуля трансформации данных для более сложных сценариев преобразования;
- разработка дополнительных модулей для мониторинга и анализа производительности интеграционных потоков;
- интеграция с инструментами DevOps для автоматизации развертывания и управления интеграциями;
- оптимизация системы для работы с большими объемами данных (Big Data) и реализация поддержки механизмов параллельной обработки для повышения производительности интеграционных потоков с хранилищами данных;
- разработка инструментов для автоматического профилирования данных и выявления аномалий в интеграционных потоках, что позволит улучшить качество данных, передаваемых в хранилища данных.
- исследование и внедрение механизмов автоматической оптимизации интеграционных потоков на основе машинного обучения, что позволит системе самообучаться и адаптироваться к изменяющимся требованиям и характеристикам источников и приемников данных.

СПИСОК ИСТОЧНИКОВ

1. Linthicum, D. S. Enterprise Application Integration. Addison-Wesley Professional. 2004. URL: <https://www.oreilly.com/library/view/enterprise-application-integration/0201615835/> [Accessed 03 January 2025].
2. Inmon W. H. Building the Data Warehouse. Wiley J. & Sons, 2005. ISBN 0471774235, 9780471774235. URL: <https://fit.hcmute.edu.vn/Resources/Docs/SubDomain/fit/ThayTuan/DataWH/Bulding%20the%20Data%20Warehouse> [Accessed 14 January 2025].
3. Koh S. C., Gunasekaran A., Goodman S. Drivers, barriers and critical success factors for ERP implementation in SMEs // Journal of Enterprise Information Management. 2011. No. 24(2). P. 204–220.
4. Trkman P. The critical success factors of business process management // International Journal of Information Management. 2010. No. 30(2). P. 125–134.
5. Melville, N. P., Kraemer K. L. & Ramasubbu N. Information technology and organizational performance: An integrative model of IT business value // MIS Quarterly. 2004. No. 28(2). P. 283–322.
6. Шереметьев И. И. Интегрированные информационные системы: учебное пособие. М.: Академия Естествознания, 2010. 235 с.
7. Kreger S., Butek R., Chance G., Ferris C., & Newcomer E. Core J2EE Patterns: Best Practices and Design Strategies. Prentice Hall. 2008. URL: <https://lira.epac.to/DOCS-TECH/Programming/Java/J2EE/Core%20J2EE%20Patterns.pdf> [Accessed 05 January 2025].
8. Ikenna Nwaiwu. Automating API Delivery. APIOps with OpenAPI. Manning Publications. 2024. URL: <https://www.manning.com/books/automating-api-delivery> [Accessed 22 January 2025].
9. Serodio, R. Serverless Architectures on AWS. Manning Publications. 2022. URL: <https://www.manning.com/books/serverless-architectures-on-aws> [Accessed 18 Desember 2024].
10. Inmon W. H. Building the Data Warehouse (4th ed.). John Wiley & Sons. 2005. URL: <https://www.wiley.com/en-us/Building+the+Data+Warehouse%2C+4th+Edition-p-9780764599446> [Accessed 12 Desember 2024].

REFERENCES

1. Linthicum DS. Enterprise Application Integration. Addison-Wesley Professional. 2004. Available from: <https://www.oreilly.com/library/view/enterprise-application-integration/0201615835/> / [Accessed 03 January 2025].

2. Inmon WH. Building the Data Warehouse. John Wiley & Sons; 2005. ISBN 0471774235, 9780471774235. Available from: <https://fit.hcmute.edu.vn/Resources/Docs/SubDomain/fit/ThayTuan/DataWH/Bulding%20the%20Data%20Warehouse> [Accessed 14 January 2025]
3. Koh SC, Gunasekaran A, & Goodman S. Drivers, barriers and critical success factors for ERP implementation in SMEs. *Journal of Enterprise Information Management*. 2011;24(2):204-220.
4. Trkman P. The critical success factors of business process management. *International Journal of Information Management*. 2010;30(2):125-134.
5. Melville NP, Kraemer KL, & Ramasubbu N. Information technology and organizational performance: An integrative model of IT business value. *MIS Quarterly*, 2004;28(2):283-322.
6. Sheremetyev II. Integrated information systems. Textbook. Moscow: Academy of Natural Sciences, 2010. 235 p. (In Russ.).
7. Kreger S, Butek R, Chance G, Ferris C, & Newcomer E. Core J2EE Patterns: Best Practices and Design Strategies. Prentice Hall. 2008. Available from: <https://lira.epac.to/DOCS-TECH/Programming/Java/J2EE/Core%20J2EE%20Patterns.pdf> [Accessed 05 January 2025].
8. Ikenna Nwaiwu. Automating API Delivery. APIOps with OpenAPI. Manning Publications. 2024. Available from: <https://www.manning.com/books/automating-api-delivery> [Accessed 22 December 2024].
9. Serodio R. Serverless Architectures on AWS. Manning Publications. 2022. Available from: <https://www.manning.com/books/serverless-architectures-on-aws> [Accessed 18 December 2024].
10. Inmon WH. Building the Data Warehouse (4th ed.). John Wiley & Sons; 2005. Available from: <https://www.wiley.com/en-us/Building+the+Data+Warehouse%2C+4th+Edition-p-9780764599446> [Accessed 12 December 2024].

ИНФОРМАЦИЯ ОБ АВТОРАХ

Анна Юрьевна Орлова, кандидат экономических наук, доцент, доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии, Северо-Кавказский федеральный университет, ResearcherID Web of Science G-3498-2018.

Анатолий Александрович Сорокин, кандидат экономических наук, доцент, доцент кафедры инжиниринга IT-решений экономического факультета, Ставропольский государственный аграрный университет, ResearcherID Web of Science MTA-2881-2025.

ВКЛАД АВТОРОВ

Анна Юрьевна Орлова. Проведение исследования – сбор, интерпретация и анализ полученных данных, визуальное моделирование. Утверждение окончательного варианта – принятие ответственности за все аспекты работы, целостность всех частей статьи и ее окончательный вариант.

Анатолий Александрович Сорокин. Проведение исследования – создание базовых классов, от которых впоследствии наследовались классы для конкретных типов источника. Подготовка и редактирование текста, составление черновика рукописи и формирование его окончательного варианта, участие в научном дизайне.

INFORMATION ABOUT THE AUTHORS

Anna Yu. Orlova – Cand. Sci. (Econ), Associate Professor, Associate Professor of the Department of Digital, Robotic Systems and Electronics at the Institute of Advanced Engineering, North-Caucasus Federal University, ResearcherID Web of Science G-3498-2018.

Anatoly A. Sorokin – Cand. Sci. (Econ), Associate Professor, Associate Professor of IT Solutions Engineering Department of the Faculty of Economics, Stavropol State Agrarian University, ResearcherID Web of Science MTA-2881-2025.

CONTRIBUTION OF THE AUTHORS

Anna Yu. Orlova. Conducting research is the collection, interpretation and analysis of the data obtained, visual modeling. The approval of the final version is the acceptance of responsibility for all aspects of the work, the integrity of all parts of the article and its final version.

Anatoly A. Sorokin. Conducting research is the creation of base classes, from which classes for specific source types were subsequently inherited. Preparation and editing of the text, drafting of the manuscript and formation of its final version, participation in scientific design.