

Котов Денис Юрьевич

*магистрант, Московский технический университет связи и информатики
(МТУСИ), научная специальность 2.3 — информационные технологии и
телекоммуникации*

ТРАНСФОРМАЦИЯ РОЛИ БИЗНЕС-АНАЛИТИКА В ЭПОХУ LOW-CODE/NO-CODE

Аннотация. В статье исследуется трансформация профессиональной роли бизнес-аналитика под влиянием распространения технологий Low-code/No-code. Цель исследования заключается в определении новых функций, компетенций и зон ответственности бизнес-аналитика в условиях сокращения объема традиционной ручной разработки программного обеспечения. Рассмотрены теоретические основы Low-code/No-code, особенности citizen development и изменения жизненного цикла создания цифровых решений. На основе анализа современных научных публикаций, практических подходов организаций и результатов систематических исследований выделены основные направления изменения деятельности бизнес-аналитика. Показано, что бизнес-аналитик постепенно переходит от преимущественно документирующей роли к роли архитектора бизнес-решений, фасилитатора взаимодействия бизнеса и информационных технологий, владельца требований и участника управления цифровой платформой. Предложена модель трансформации компетенций бизнес-аналитика, включающая аналитические, технологические, управленческие и коммуникационные компоненты. Определены преимущества и риски использования Low-code/No-code для аналитической деятельности, а также сформулированы рекомендации по развитию компетенций специалистов.

Ключевые слова: бизнес-аналитик, Low-code, No-code, цифровая трансформация, citizen development, бизнес-процессы, цифровые платформы, управление требованиями, автоматизация, информационные системы.

Abstract. The article examines the transformation of the professional role of the business analyst under the influence of Low-code/No-code technologies. The purpose of the study is to identify new functions, competencies and areas of responsibility of the business analyst in an environment where the amount of traditional manual software development is decreasing. The paper considers the theoretical foundations of Low-code/No-code, citizen development and changes in the life cycle of digital solution development. Based on an analysis of contemporary academic publications, organizational practices and systematic reviews, the main directions of change in business analysis are identified. It is shown that the business analyst is gradually moving from a predominantly documenting role to the role of a business-solution architect, facilitator of interaction between business and IT, requirements owner and participant in digital platform governance. A competency transformation model is proposed that combines analytical, technological, managerial and communication components. The advantages and risks of Low-code/No-code for business analysis are systematized, and recommendations for professional development are formulated. The study concludes that the value of the business analyst increasingly depends on the ability to transform a business problem into a manageable, technologically feasible and economically justified digital solution.

Keywords: business analyst, Low-code, No-code, digital transformation, citizen development, business processes, digital platforms, requirements management, automation, information systems.

Введение

Переход к цифровым моделям ведения бизнеса меняет процесс создания информационных систем. Последовательная работа заказчика, аналитика, архитектора, программиста, тестировщика заменяется интеграционными платформами, что сокращает путь от постановки задачи до готового продукта. Ключевую роль в этом играют технологии Low-code/No-code, позволяющие бизнесу самостоятельно запускать новые сервисы, объединять системы и

строить интерфейсы с минимальным привлечением классических разработчиков.

Актуальность темы обусловлена изменением инструментария программистов, а также организационной модели цифрового производства в целом. В частности, меняется содержание работы бизнес-аналитика – специалиста, традиционно выступающего связующим звеном между представителями бизнеса и техническими командами. Если ранее основная часть его деятельности была связана с выявлением, структурированием, документированием и согласованием требований, то в современной цифровой среде бизнес-аналитик получает возможность непосредственно участвовать в создании прототипов, настройке автоматизированных процессов, моделировании пользовательских сценариев, проверке гипотез и даже самостоятельной реализации относительно простых цифровых решений.

Научная проблема заключается в необходимости определить, каким образом изменение технологической среды влияет на профессиональную идентичность бизнес-аналитика, какие традиционные функции сохраняются, какие трансформируются, а какие появляются впервые. Дополнительную сложность создает то обстоятельство, что Low-code/No-code не устраняет необходимость анализа требований. Напротив, увеличение количества пользователей, способных самостоятельно создавать цифровые решения, повышает требования к качеству постановки задач, управлению архитектурными ограничениями, контролю данных, безопасности и интеграции.

Современные исследования подтверждают значимость данной проблематики. Систематический обзор литературы, посвященный внедрению Low-code/No-code и citizen development, выявил среди основных преимуществ процессное совершенствование, развитие цифровых навыков, повышение инновационной активности, возможность кастомизации и интеграции решений. Одновременно среди проблем отмечаются фрагментация платформ, зависимость от поставщика, ограничения масштабируемости, риски безопасности, проблемы

it-управления, дефицит знаний и конфликты ролей [1; 3; 8–12]. При этом, аналитические модели становятся ближе к исполняемому коду, а базовой рамкой для профессионального анализа остается BABOK Guide, связывающий работу с требованиями и стейкхолдерами с созданием ценности.

Также были проанализированы организации, внедряющие стратегии Low-code/No-code. Авторы показали, что эффективность такого подхода определяется не только технологией, но и организационными механизмами, включая вопросы управления, безопасности, соответствия требованиям и распределения ответственности [2].

Таким образом, задача современного бизнес-аналитика заключается уже не только в формулировании требований к информационной системе. Он должен понимать возможности и ограничения цифровой платформы, оценивать реализуемость предлагаемых решений, участвовать в создании прототипов и обеспечивать соответствие создаваемых приложений целям организации.

1. Концептуальные основы Low-code/No-code и факторы, определяющие трансформацию роли бизнес-аналитика

Подход Low-code/No-code базируется на использовании платформ для создания ИТ-решений посредством визуального проектирования и параметрической настройки. Ключевое отличие от традиционной инженерии программного обеспечения заключается в замещении императивного написания исходного кода автоматизированной сборкой из предопределенных компонентов и встроенными инструментами интеграции. Если Low-code предполагает, что пользователь способен разобраться в архитектуре данных и интеграционных потоках, то No-code полностью скрывает техническую сложность за простым интерфейсом, позволяя создавать решения с минимальными навыками разработки.

Актуальные научные работы свидетельствуют о рассматриваемых рассматриваемых подходах, что связано с развитием инструментария в направлении графического интерфейса, декларативной спецификации сценариев

и применения генеративных моделей [3]. В связи с чем, использование технологий искусственного интеллекта дополнительно актуализирует проблематику управления, вызванную в том числе технологическими угрозами [15].

Принципиальным организационным сдвигом, который несут технологии Low-code/No-code, становится переход к концепции citizen development, в соответствии с которой разработку цифровых инструментов берут на себя сотрудники профильных отделов, а не профессиональные программисты. Такой подход позволяет передавать часть задач непосредственно в бизнес-подразделения, сокращая время между обнаружением проблемы и созданием ее цифрового решения.

Однако демократизация разработки программного обеспечения не означает устранения профессиональных специалистов. Напротив, исследования показывают, что внедрение Low-code/No-code требует новых механизмов управления, поскольку увеличение числа разработчиков внутри организации потенциально повышает риски появления дублирующих приложений, неуправляемых интеграций, нарушений безопасности и так называемого shadow IT [2].

В этом контексте особое значение приобретает бизнес-аналитик. С одной стороны, он обладает пониманием бизнес-процессов и целей организации, с другой, должен понимать технологические возможности цифровой платформы. Поэтому именно бизнес-аналитик способен стать одним из основных участников перехода от традиционной модели «бизнес формулирует требования – IT реализует» к модели совместного проектирования цифровых решений.

Традиционный бизнес-анализ включает несколько взаимосвязанных функций. Данная совокупность функций соответствует общему представлению о профессиональной практике бизнес-анализа [8].

- выявление заинтересованных сторон;
- сбор и анализ требований;

- моделирование бизнес-процессов;
- формирование функциональных и нефункциональных требований;
- согласование требований;
- подготовку спецификаций;
- участие в приемочном тестировании;
- анализ результатов внедрения.

При этом, визуальное моделирование процессов обеспечивает сближение аналитической схемы, проектной документации и исполняемого кода. Вследствие этого граница между этапом проектирования и программной реализацией нивелируется, что выступает одним из ключевых условий трансформации роли бизнес-аналитика.

2. Анализ трансформации роли бизнес-аналитика

Традиционная разработка строится на длинной цепочке посредников: аналитик пишет документ, архитектор проектирует структуру, разработчик воплощает её в коде. Эта формальность гарантирует контроль над процессом, но порождает две системные проблемы: существенное замедление вывода решения на рынок и неизбежный разрыв между первоначальным запросом бизнеса и итоговым функционалом.

Технологии Low-code/No-code минимизируют количество промежуточных этапов трансляции требований, что позволяет бизнес-аналитику осуществлять переход от текстового описания к интерактивному прототипу синхронно со сбором и валидацией потребностей. В отличие от традиционной каскадной модели (бизнес-проблема → документация → проектирование → реализация), подход Low-code/No-code базируется на сокращенном итеративном цикле (проблема → визуальный макет → обратная связь → настройка). В данной парадигме прототип объединяет функции как средства анализа требований, так и основы для программной реализации. Ключевым следствием становится смена профессиональной парадигмы: аналитик эволюционирует от роли пассивного

фиксатора спецификаций к позиции активного участника конфигурирования цифрового решения.

Подобная трансформация позволяет сократить разрыв в понимании задачи между бизнесом и разработкой. Визуальный прототип позволяет пользователю увидеть предполагаемое решение еще до завершения разработки. Если обнаруживается несоответствие, оно устраняется на раннем этапе. Исследования моделирования в Low-code подтверждают сближение моделей и реализации цифрового решения [11].

Исследования Low-code показывают, что одним из существенных преимуществ такого подхода является ускорение разработки и повышение взаимодействия между бизнес-подразделениями и IT-специалистами [3]. При этом результаты систематического исследования 2026 года показывают, что положительный эффект по скорости разработки имеет достаточно устойчивую исследовательскую поддержку, тогда как влияние на качество и сложность зависит от контекста применения [4].

Следовательно, нельзя рассматривать Low-code/No-code как универсальное средство повышения эффективности. Его преимущества реализуются при правильном выборе класса задач, наличии управления платформой и соответствующем уровне компетенций пользователей.

Одним из главных результатов проведенного анализа является вывод о переходе бизнес-аналитика от роли «переводчика требований» к роли «архитектора бизнес-решения». Такую трансформацию можно представить через несколько этапов.

Первый этап – от сбора требований к совместному проектированию. Традиционная деятельность бизнес-аналитика начинается с интервью, изучения документов, наблюдения за процессами и формирования требований. В Low-code/No-code-среде этот этап дополняется интерактивным прототипированием.

Так у аналитика появляется возможность прямой демонстрации исполняемого сценария заказчику. На примере автоматизации согласования

заявки это выражается в замене текстовой спецификации десятков требований на конфигурируемую визуальную модель: маршрутизацию, матрицу ролей, триггеры переходов и системные уведомления. В результате предмет обсуждения смещается от абстрактной технической документации к конкретному цифровому прототипу. В связи с чем возрастает уровень обратной связи, что способствует выявлению латентных бизнес-потребностей.

Второй этап – от простого описания задач к непосредственному конструированию логики. Документация не исчезает, но ее роль трансформируется: вместо статичных текстов основным инструментом становятся живые схемы процессов, настроенные алгоритмы и работающие прототипы.

Компетенции бизнес-аналитика включают определение границ между ручными и автоматизированными операциями, спецификацию потоков данных, триггеров запуска, правил маршрутизации, сценариев обработки исключений, матрицы ролевого доступа и необходимых межсистемных интеграций. Таким образом, аналитик начинает работать не только с требованиями, но и с исполняемой бизнес-логикой.

Третий этап – от взаимодействия с разработчиками к взаимодействию с платформой. Еще одна важная трансформация связана с тем, что Low-code-платформа становится своеобразным «технологическим партнером» аналитика. Бизнес-аналитик должен понимать возможности конкретной платформы:

- типы данных;
- способы моделирования процессов;
- компоненты пользовательского интерфейса;
- механизмы автоматизации;
- интеграционные возможности;
- ограничения API;
- механизмы авторизации;
- управление ролями;

- средства мониторинга;
- инструменты тестирования;
- механизмы публикации и версионирования.

Это не превращает бизнес-аналитика в профессионального программиста. Однако граница между бизнесовой и технической компетентностью становится значительно более проницаемой.

Четвертый этап – от требований к ценности. В традиционном подходе результат работы аналитика часто оценивается по полноте и качеству документации. В Low-code/No-code-среде более важным становится конечный эффект.

Аналитик должен отвечать на вопросы:

- какую проблему решает приложение;
- сколько времени экономится;
- какие операции исключаются;
- насколько снижается количество ошибок;
- какие пользователи получают выгоду;
- какие показатели процесса изменяются;
- насколько решение масштабируемо.

Таким образом, происходит переход от requirement-oriented к value-oriented бизнес-анализу.

Трансформация роли приводит к необходимости расширения компетентностной модели специалиста. Предлагается выделить четыре группы компетенций: аналитические, технологические, управленческие и коммуникативные.

Первая группа – аналитические компетенции, к которым относятся системное мышление, процессный анализ, моделирование бизнес-процессов, выявление проблем, формирование требований, анализ данных, управление бизнес-правилами, оценка эффективности процессов. Эта группа остается

фундаментальной. Даже самая совершенная Low-code-платформа не способна заменить качественное понимание бизнеса.

Вторая группа – технологические компетенции. Бизнес-аналитик должен понимать принципы работы Low-code/No-code, архитектуру платформ, основы интеграции, API, базы данных, управление доступом, информационную безопасность, принципы тестирования, управление версиями, ограничения масштабируемости. Важен не столько навык написания большого количества кода, сколько способность понимать технологические последствия принимаемых решений.

Третья группа – управленческие компетенции. Рост citizen development требует участия бизнес-аналитика в управлении цифровыми инициативами. Специалист должен уметь приоритизировать инициативы, оценивать риски, определять критерии готовности, организовывать взаимодействие подразделений, участвовать в формировании стандартов, контролировать жизненный цикл приложений, определять необходимость привлечения IT, оценивать экономический эффект. Таким образом, бизнес-аналитик частично приближается к функциям product owner и координатора цифровой трансформации.

Четвертая группа – коммуникационные компетенции. В условиях сокращения дистанции между бизнесом и IT особенно важными становятся навыки фасилитации. Бизнес-аналитик должен уметь организовывать совместную работу пользователей, разработчиков, владельцев процессов, специалистов по информационной безопасности и руководителей. Его задача состоит не просто в передаче информации между сторонами, а в формировании общего понимания проблемы и согласовании цифрового решения.

Распространение Low-code/No-code имеет двойственное влияние на качество аналитической работы.

С одной стороны, повышается наглядность требований. Пользователь может взаимодействовать с прототипом, а не только читать текстовую

спецификацию. Сокращается количество коммуникационных циклов. Ошибки могут обнаруживаться раньше.

С другой стороны, существует риск преждевременного перехода от анализа проблемы к реализации интерфейса.

Пользователь может сформулировать решение в виде фразы «Нам нужно приложение с такими-то полями», хотя реальная проблема заключается в неэффективности самого бизнес-процесса. Поэтому бизнес-аналитик должен сохранять принцип «сначала проблема и бизнес-ценность, затем цифровое решение».

Именно здесь проявляется отличие профессионального аналитика от обычного citizen developer.

Citizen developer может создать приложение. Бизнес-аналитик должен определить, нужно ли вообще создавать это приложение, какую проблему оно решает, какие процессы должны измениться и какой результат будет считаться успешным.

Несмотря на значительные преимущества, переход к Low-code/No-code создает следующие риски.

1. Технологический долг. Быстро созданные приложения могут оказаться недостаточно документированными, зависеть от конкретной платформы или отдельных пользователей. По мере увеличения числа приложений возрастает стоимость их сопровождения.

2. Shadow IT. Самостоятельное создание решений пользователями может привести к появлению информационных систем, не учтенных в корпоративном ИТ-ландшафте. Особую опасность представляет использование несанкционированных источников данных и интеграций.

3. Проблемы информационной безопасности. Даже если платформа обеспечивает базовые механизмы защиты, ошибки настройки ролей и доступа могут привести к раскрытию информации. Поэтому безопасность должна учитываться уже на этапе бизнес-анализа.

4. Зависимость от платформы. Организация может оказаться зависимой от конкретного поставщика Low-code/No-code. Необходимо учитывать стоимость лицензирования, возможности экспорта данных, переносимость решений, доступность API, перспективы развития платформы, условия изменения тарифов.

5. Снижение качества архитектуры. Простота создания приложения может создать иллюзию простоты самого решения. Если аналитик не учитывает архитектурные ограничения, приложение может работать на уровне небольшого подразделения, но оказаться непригодным для масштабирования.

Систематические исследования подтверждают наличие подобных проблем: среди барьеров Low-code/No-code называются фрагментация платформ, vendor lock-in, недостаточная масштабируемость, проблемы архитектуры, безопасности и управления [1].

Одним из перспективных направлений развития профессии становится участие бизнес-аналитика в платформенном управлении. Если организация использует десятки или сотни Low-code-приложений, возникает необходимость централизованного учета владельцев приложений, назначения каждого приложения, используемых данных, интеграций, пользователей, критичности, требований безопасности, сроков сопровождения.

Бизнес-аналитик может участвовать в разработке критериев классификации приложений. Например, можно выделить три уровня:

Уровень 1 – локальные решения. Используются небольшой группой сотрудников и не работают с критичными данными.

Уровень 2 – корпоративные решения. Используются несколькими подразделениями и интегрируются с корпоративными системами.

Уровень 3 – критические решения. Влияют на ключевые бизнес-процессы, используют критичные данные или являются частью внешнего взаимодействия организации. Для третьего уровня должна применяться более строгая

архитектурная и ИТ-проверка. Подобная модель позволяет совместить преимущества citizen development с требованиями корпоративного управления.

Развитие Low-code/No-code постепенно сопровождается интеграцией искусственного интеллекта. Современные платформы начинают использовать AI для генерации элементов приложения, преобразования естественно-языковых запросов в бизнес-логику, создания автоматизированных процессов и подготовки компонентов пользовательского интерфейса. При использовании ИИ дополнительно актуализируются вопросы управления соответствующими системными рисками [15].

Это означает переход от визуального программирования к более высокому уровню абстракции: пользователь описывает намерение, а платформа предлагает технологическую реализацию. В исследовательской литературе уже отмечается движение от instruction-based к intent-based development, усиливаемое развитием agentic AI [3].

Для бизнес-аналитика это означает дальнейшее усиление его роли. Если платформа становится способной автоматически создавать приложение на основании текстового описания, критическим фактором становится качество самого описания бизнес-проблемы. Иными словами, ценность специалиста смещается от способности «сказать системе, как что-то сделать» к способности определить:

- что именно нужно сделать;
- зачем это нужно;
- какие ограничения существуют;
- какие данные допустимо использовать;
- какие последствия возникнут;
- каким образом оценивать результат.

В перспективе бизнес-аналитик может выступать своеобразным «архитектором намерений», формируя требования на уровне целей, бизнес-правил, ограничений и критериев качества.

3. Анализ результатов и перспективы развития профессии бизнес-аналитика

Для систематизации результатов исследования целесообразно сопоставить традиционную и трансформированную модели деятельности бизнес-аналитика. Таблица 1. Сопоставление традиционной и трансформированной моделей деятельности бизнес-аналитика

Область деятельности	Традиционный подход	Low-code/No-code подход
Работа с требованиями	Документирование	Интерактивное моделирование
Прототипирование	Часто выполняется отдельно	Является частью анализа
Взаимодействие с ИТ	Передача требований	Совместное проектирование
Работа с процессами	Описание текущего и целевого состояния	Непосредственная настройка автоматизации
Технологические знания	Базовые	Расширенные платформенные
Оценка результата	Соответствие требованиям	Соответствие требованиям + бизнес-эффект
Участие в разработке	Косвенное	Частично непосредственное
Управление рисками	В основном на уровне требований	Также на уровне платформы и данных
Работа с citizen developers	Не является обязательной	Может стать одной из основных функций
Основной результат	Спецификация	Рабочее цифровое решение и модель его ценности

Анализ показывает, что трансформация не означает исчезновения классического бизнес-анализа. Напротив, базовые аналитические компетенции приобретают еще большую ценность. Изменяется главным образом форма их реализации.

Результаты систематических исследований позволяют выделить несколько закономерностей. Во-первых, Low-code/No-code действительно снижает технологический барьер создания приложений [1; 3]. Во-вторых, скорость разработки является одним из наиболее устойчиво подтверждаемых преимуществ Low-code, однако влияние технологии на качество и сложность решения зависит от контекста [4]. В-третьих, успех внедрения определяется не только платформой, но и организационными механизмами управления, распределением ролей и развитием цифровых компетенций [2; 5]. В-четвертых, по мере роста масштабов использования Low-code увеличивается потребность в управлении архитектурой, безопасностью, данными и жизненным циклом приложений. Современные корпоративные платформы уже рассматриваются не только как инструменты разработки, но и как среды, включающие механизмы управления, безопасности, интеграции и эксплуатации [6]. Следовательно, трансформация бизнес-аналитика имеет не линейный, а многоуровневый характер. Поэтому оценка качества должна опираться на совокупность характеристик цифрового продукта [14].

Для организаций, внедряющих Low-code/No-code, целесообразно сформировать программу развития компетенций бизнес-аналитиков.

На первом этапе специалисту необходимо освоить основы конкретной Low-code-платформы. При этом задача не должна сводиться к подготовке профессионального разработчика. Главная цель – понимание технологических возможностей и ограничений.

На втором этапе необходимо развивать навыки прототипирования. Аналитик должен уметь быстро создавать минимальные версии цифровых процессов и использовать их для проверки требований.

На третьем этапе следует усилить компетенции в области данных. Современное цифровое решение практически всегда связано с несколькими источниками информации, поэтому аналитик должен понимать структуру данных, качество данных, права доступа и принципы интеграции.

На четвертом этапе необходимо освоить основы платформенного управления: жизненный цикл приложений, управление версиями, разграничение доступа, контроль публикации и документирование.

На пятом этапе необходимо сформировать компетенции оценки бизнес-эффекта.

Можно использовать следующую систему показателей:

- сокращение времени выполнения процесса;
- снижение количества ручных операций;
- снижение количества ошибок;
- сокращение времени обработки заявки;
- уменьшение нагрузки на сотрудников;
- увеличение прозрачности процесса;
- повышение удовлетворенности пользователей;
- снижение стоимости операции.

Таким образом, эффективность Low-code-решения следует измерять не количеством созданных приложений, а изменением бизнес-показателей.

В среднесрочной перспективе можно выделить несколько направлений развития профессии.

Первое направление – business technologist. Бизнес-аналитик будет обладать достаточным уровнем технологической компетентности для самостоятельного создания прототипов и простых цифровых решений.

Второе направление – product-oriented analyst. Специалист будет концентрироваться не на отдельных требованиях, а на жизненном цикле цифрового продукта и создаваемой им ценности.

Третье направление – platform analyst. Бизнес-аналитик будет участвовать в управлении корпоративной Low-code-платформой, стандартах разработки и классификации приложений.

Четвертое направление – AI-enabled analyst. Использование искусственного интеллекта позволит автоматизировать часть рутинных

аналитических задач: формирование черновиков требований, классификацию обратной связи, подготовку вариантов процессов, генерацию прототипов и анализ данных. При использовании ИИ дополнительно актуализируются вопросы управления соответствующими системными рисками [15].

Пятое направление – enterprise process architect. По мере увеличения числа приложений возрастет значение специалистов, способных видеть не отдельную автоматизированную операцию, а весь цифровой ландшафт организации.

Таким образом, профессия не исчезает под влиянием Low-code/No-code, а становится более технологичной и одновременно более ориентированной на бизнес-ценность.

Заключение

Проведенное исследование показывает, что распространение Low-code/No-code является не просто изменением инструментария разработки программного обеспечения, а фактором структурной трансформации бизнес-анализа.

Основным результатом становится изменение положения бизнес-аналитика в жизненном цикле цифрового решения. Ключевой компетенцией становится синтез бизнесового, процессного и технологического мышления, понимание не только нужд пользователей, но и ограничений платформы, данных и безопасности.

При этом, развитие citizen development делает аналитика ключевым посредником между бизнесом, самостоятельно создающим приложения, и профессиональными IT-командами.

Low-code/No-code не заменяет анализ, а повышает его ценность. Если технологический барьер снижается, то возрастает вероятность создания большого количества приложений, которые сами по себе не гарантируют повышения эффективности. Поэтому бизнес-аналитик должен концентрироваться на вопросах ценности: какую проблему решает решение,

почему ее следует автоматизировать, каким будет экономический и организационный эффект и какие риски возникают.

Исследование также позволяет сделать вывод о необходимости изменения образовательной модели подготовки бизнес-аналитиков. Помимо классических дисциплин, связанных с управлением требованиями, моделированием процессов и системным анализом, специалистам необходимы знания Low-code/No-code, архитектуры приложений, API, данных, информационной безопасности, платформенного управления и искусственного интеллекта. При использовании ИИ дополнительно актуализируются вопросы управления соответствующими системными рисками [15].

Практическая значимость предложенной модели заключается в возможности использовать ее при разработке профилей должностей, программ повышения квалификации, корпоративных стандартов citizen development и критериев оценки цифровых компетенций.

Перспективным направлением дальнейших исследований является разработка количественной методики оценки эффективности бизнес-аналитика в Low-code/No-code-среде. Такая методика может учитывать скорость создания прототипа, количество итераций согласования, степень соответствия решения требованиям, экономический эффект, качество данных, количество дефектов и уровень пользовательской удовлетворенности.

Таким образом, трансформация роли бизнес-аналитика должна рассматриваться не как вытеснение специалиста автоматизированными средствами, а как переход к более высокому уровню профессиональной деятельности. В эпоху Low-code/No-code ценность бизнес-аналитика все в большей степени определяется не способностью детально описать будущую систему, а способностью превратить бизнес-проблему в управляемое, технологически реализуемое и экономически обоснованное цифровое решение.

СПИСОК ИСТОЧНИКОВ

1. Ajimati M. O., Carroll N., Maher M. Adoption of low-code and no-code development: A systematic literature review and future research agenda // Journal of Systems and Software. – 2025. – Vol. 222. – Article 112300. – DOI: 10.1016/j.jss.2024.112300.
2. Binzer B., Elshan E., Fürstenau D., Winkler T. J. Establishing a Low-Code/No-Code-Enabled Citizen Development Strategy // MIS Quarterly Executive. – 2024. – Vol. 23, No. 3. – P. 253–273. – DOI: 10.17705/2msqe.00097.
3. Rokis K., Kirikova M. Exploring Low-Code Development: A Comprehensive Literature Review // Complex Systems Informatics and Modeling Quarterly. – 2023. – No. 36. – P. 68–86. – DOI: 10.7250/csimq.2023-36.04.
4. Gao D., Fagerholm F., Toivanen V. What does current research say about the viability of low-code development? A systematic literature review // Journal of Systems and Software. – 2026. – Vol. 239. – Article 112893.
5. Prinz N., Huber M. Citizen Development als Treiber der digitalen Transformation – Aktuelle Ansätze bei der Adoption von Low-Code Development Plattformen // HMD Praxis der Wirtschaftsinformatik. – 2024. – Vol. 61. – P. 1115–1135. – DOI: 10.1365/s40702-023-01021-8.
6. Gartner. Critical Capabilities for Enterprise Low-Code Application Platforms. – 2025.
7. El Kamouchi H., Kissi M., El Beggar O. Low-code/No-code Development: A systematic literature review // SITA 2023. – 2023. – P. 1–8. – DOI: 10.1109/SITA60746.2023.10373712.
8. International Institute of Business Analysis. A Guide to the Business Analysis Body of Knowledge (BABOK Guide). – Version 3. – Toronto: IIBA, 2015.
9. Bock A. C., Frank U. Low-Code Platform // Business & Information Systems Engineering. – 2021. – Vol. 63, No. 6. – P. 733–740. – DOI: 10.1007/s12599-021-00726-8.

10. Waszkowski R. Low-code platform for automating business processes in manufacturing // IFAC-PapersOnLine. – 2019. – Vol. 52, No. 10. – P. 376–381. – DOI: 10.1016/j.ifacol.2019.10.060.
11. Bucaioni A., Cicchetti A., Ciccozzi F. Modelling in low-code development: a multi-vocal systematic review // Software and Systems Modeling. – 2022. – Vol. 21. – P. 1959–1981. – DOI: 10.1007/s10270-021-00964-0.
12. Lázaro J., Brambilla M., Cabot J. What about the usability in low-code platforms? A systematic literature review // Computer Languages, Systems & Structures. – 2023. – Vol. 74. – Article 101185. – DOI: 10.1016/j.cola.2022.101185.
13. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. – Geneva: ISO, 2022.
14. ISO/IEC 25010:2023. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model. – Geneva: ISO, 2023.
15. ISO/IEC 42001:2023. Information technology — Artificial intelligence — Management system. – Geneva: ISO, 2023.