

УДК 004.4

## **ВЛИЯНИЕ LOW-CODE И NO-CODE ПЛАТФОРМ НА ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

А.М. ЗЕНЕВИЧ<sup>1</sup>, Д.В. КНЫШ<sup>2</sup>, Е.Г. РУБЕЛЬ<sup>2</sup>

<sup>1</sup>к.э.н., доцент кафедры экономической информатики,

<sup>2</sup>студенты учебной группы 22ДЦИ2

Белорусский государственный экономический университет

г. Минск, Республика Беларусь

*Аннотация. В статье исследуется влияние low-code и no-code платформ на процесс разработки программного обеспечения. Рассматриваются преимущества данных технологий, такие как ускорение разработки, снижение затрат и демократизация создания приложений, а также анализируются их недостатки, включая ограничения масштабируемости, гибкости и проблемы vendor lock-in. Особое внимание уделяется роли этих платформ в контексте профессиональной разработки и бизнес-операций, а также их влиянию на рынок труда для начинающих разработчиков. На основе анализа делается вывод о необходимости гибридного подхода, сочетающего low-code/no-code решения с традиционным программированием для достижения оптимальных результатов.*

*Ключевые слова: low-code, no-code, разработка программного обеспечения, цифровая трансформация, автоматизация, vendor lock-in, гибридный подход.*

## **THE IMPACT OF LOW-CODE AND NO-CODE PLATFORMS ON THE SOFTWARE DEVELOPMENT PROCESS**

A.M. ZENEVICH<sup>2</sup>, D.V. KNYSH<sup>2</sup>, E.G. RUBEL<sup>2</sup>,

<sup>1</sup>Associate Professor of the Department of Economic Informatics, Candidate of Economic Sciences

<sup>2</sup>group students 22DDI2

Belarusian State Economic University

Minsk, Republic of Belarus

*Abstract. The article examines the impact of low-code and no-code platforms on the software development process. The advantages of these*

*technologies, such as accelerated development, reduced costs, and democratization of application creation, are discussed, along with their disadvantages, including scalability limitations, flexibility issues, and vendor lock-in problems. Special attention is paid to the role of these platforms in the context of professional development and business operations, as well as their impact on the job market for junior developers. The analysis concludes with the necessity of a hybrid approach that combines low-code/no-code solutions with traditional programming to achieve optimal results.*

*Key words: low-code, no-code, software development, digital transformation, automation, vendor lock-in, hybrid approach.*

С ускорением цифровой трансформации и ростом потребности бизнеса в быстрых программных решениях возрастает значение платформ, позволяющих автоматизировать разработку и сократить время вывода продукта на рынок. Low-code и no-code платформы (например, Airtable, Unqork, Zapier, Mendix, Bubble, Microsoft Power Apps, Salesforce Lightning) предлагают альтернативу традиционному программированию, предоставляя визуальные интерфейсы и модульные элементы, что позволяет существенно упростить и ускорить процесс создания программного обеспечения (далее ПО).

Low-Code – это подход к разработке, который минимизирует ручное написание кода, используя визуальные интерфейсы и готовые модули. No-Code идет еще дальше, позволяя создавать приложения без написания кода вообще. Эти технологии идеально подходят для бизнеса, который хочет быстро адаптироваться к изменениям и внедрять новые решения [1]. Эти платформы обещают ускорение процессов и снижение затрат, однако они также вызывают вопросы о гибкости, безопасности и возможности масштабирования приложений.

Целью работы является анализ влияния low-code и no-code платформ на процесс разработки программного обеспечения. Основной задачей является выявление преимуществ и недостатков этих платформ, а также понимание их роли в контексте профессиональной разработки ПО и бизнес-операций. Важно рассмотреть, как они изме-

няют подходы к разработке, кто становится основными пользователями таких платформ, и какие потенциальные риски возникают при их применении.

Метод low-code поддерживают разные бизнес-сервисы: например, ESM-, ECM-, BPM-, CRM-системы и другие. За счет этого компании, которые пользуются данными сервисами, могут создавать уникальный софт для автоматизации внутрикorporативных рабочих задач [2]. Это делает ПО более адаптивным и легко настраиваемым под нужды конкретного бизнеса.

Согласно аналитике Gartner и Forrester, low-code и no-code технологии способствуют демократизации разработки — то есть передаче части функций по созданию приложений от профессиональных разработчиков к бизнес-пользователям [3]. Согласно прогнозу Gartner, к 2025 году 70% новых приложений, разрабатываемых предприятиями, будут использовать технологии low-code или no-code, что способствует вовлечению бизнес-пользователей в процесс разработки. Это особенно актуально для компаний, где ИТ-отделы перегружены, а потребности бизнеса требуют оперативной реализации цифровых решений. Например, сотрудник отдела продаж может собрать простую форму CRM в *Airtable* или автоматизировать рутину с помощью *Zapier*, не обращаясь к разработчикам. Такой подход снижает порог входа и ускоряет цифровую трансформацию.

Аналитики также подчеркивают, что скорость разработки на LC/NC-платформах может быть в 3–5 раз выше, чем при использовании традиционного подхода. Это особенно важно в сферах с высокой конкуренцией и быстро меняющимися условиями. Так, например, платформа *Unqork* используется банками и страховыми компаниями для быстрого развертывания финтех-приложений и клиентских порталов, позволяя внедрять новые функции в считанные недели вместо месяцев.

Однако, несмотря на рост популярности LC/NC, у профессионального ИТ-сообщества есть серьезные опасения.

Во-первых, это технические ограничения платформ: сложность масштабирования, низкая гибкость при реализации нетривиальной логики, а также снижение производительности при высоких нагрузках.

ках. Например, приложение, созданное на Bubble, может демонстрировать задержки и замедления при большом числе одновременных пользователей или при обработке сложных запросов.

Во-вторых, возникает проблема vendor lock-in — привязки к конкретному поставщику платформы. Приложения, разработанные, например, в Salesforce Lightning, часто невозможно перенести на другую платформу без полного переписывания, что ограничивает свободу и стратегическую гибкость компании.

Также высказывается мнение, что массовое внедрение no-code инструментов может стать угрозой для профессии начинающих разработчиков, особенно на уровне junior. Поскольку такие платформы позволяют бизнес-пользователям самостоятельно разрабатывать решения без глубоких знаний программирования, перед ИТ-специалистами встает задача перейти от выполнения рутинных задач к более сложной деятельности – архитектуре систем, интеграции сервисов и обеспечению масштабируемости и безопасности решений. В связи с этим в образовательных программах возрастает значимость междисциплинарного подхода, который сочетает технические компетенции с навыками бизнес-анализа, проектного управления и взаимодействия с пользователями. Таким образом, LC/NC-платформы не отменяют необходимость в разработчиках, но меняют характер их задач, формируя спрос на универсальных специалистов нового типа.

Многие организации придерживаются гибридного подхода, при котором LC/NC не заменяют, а дополняют классическую разработку. Такие платформы применяются там, где они действительно эффективны – в автоматизации внутренних процессов, в работе с формами, в создании аналитических дашбордов. Например, в Uber с помощью Microsoft Power Apps создаются внутренние дашборды для бизнес-операций, однако основная инфраструктура компании работает на мощных backend-системах с использованием Python и Go [4].

Таким образом, low-code и no-code – это не угроза, а эволюция разработки ПО. Они открывают новые возможности для быстрой реализации идей, особенно на ранних этапах – в прототипировании, создании MVP, автоматизации бизнес-процессов. Благодаря им компании могут оперативно адаптироваться к изменениям рынка. Это особенно важно для малого и среднего бизнеса, у которого нет ресурсов на полноценную ИТ-команду.

Однако LC/NC-подход – это не «универсальный ключ». У него есть ограничения: масштабируемость, производительность, безопасность и кастомизация. Для критически важных систем или архитектурно сложных решений традиционная разработка по-прежнему незаменима.

Кроме того, LC/NC-платформы меняют роль разработчика: теперь важно не только писать код, но и уметь интегрировать готовые решения, выстраивать архитектуру вокруг API и быстро переключаться между задачами разной сложности.

В заключение следует отметить, что успешная реализация цифровых инициатив требует баланса между визуальной разработкой и классическим программированием. Только в этом случае можно эффективно использовать преимущества обеих парадигм и достигать устойчивых результатов в сфере ИТ.

## ЛИТЕРАТУРА

1. Low-code/No-code: новые подходы к разработке информационных систем. – URL: [<https://compit.by/customers/razrabotka-informatsionnyh-sistem/low-code-no-code/>] (дата обращения: 07.04.2025).

2. Обзор российских low-code платформ для бизнеса. – URL: <https://it-guild.com/info/blog/obzor-rossijskih-low-code-platform-dlya-biznesa/> (дата обращения: 07.04.2025).

3. Low-Code Development Platform Statistics and Trends // AIMultiple. – URL: <https://research.aimultiple.com/low-code-statistics/> (дата обращения: 07.04.2025).

4. Uber transforms business processes with Microsoft Power Platform // Microsoft Customer Stories. – URL: <https://www.microsoft.com/en/customers/story/19751-uber-technologies-inc-power-automate> (дата обращения: 07.04.2025).

## REFERENCES

1. Low-code/No-code: novye podhody k razrabotke informacionnyh sistem. – URL: <https://compit.by/customers/razrabotka-informatsionnyh-sistem/low-code-no-code/> (accessed: 07.04.2025).

2. Review of Russian low-code platforms for business. – URL: <https://it-guild.com/info/blog/obzor-rossijskih-low-code-platform-dlya-biznesa/> (accessed: 07.04.2025).

3. Low-Code Development Platform Statistics and Trends // AIMultiple. – URL: <https://research.aimultiple.com/low-code-statistics/> (accessed: 07.04.2025).

4. Uber transforms business processes with Microsoft Power Platform // Microsoft Customer Stories. – URL: <https://www.microsoft.com/en/customers/story/19751-uber-technologies-inc-power-automate> (accessed: 07.04.2025).