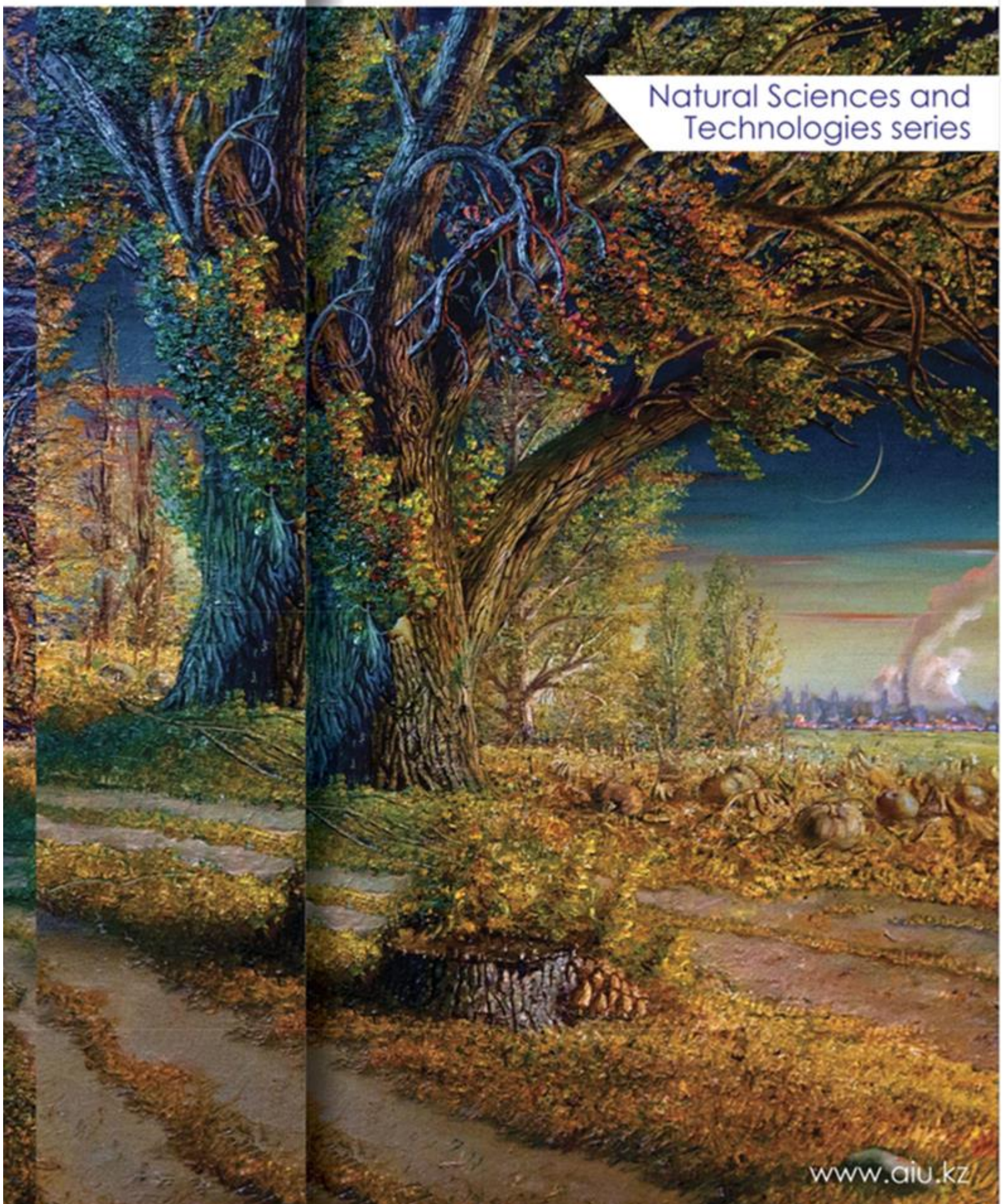


INTERNATIONAL SCIENCE REVIEWS



№3 (4) 2023

Natural Sciences and
Technologies series





INTERNATIONAL SCIENCE REVIEWS

Natural Sciences and Technologies series

Has been published since 2020

№3 (4) 2023

Astana

EDITOR-IN-CHIEF:

Doctor of Physical and Mathematical Sciences, Academician of NAS RK, Professor
Kalimoldayev M. N.

DEPUTY EDITOR-IN-CHIEF:

Doctor of Biological Sciences, Professor
Myrzagaliyeva A. B.

EDITORIAL BOARD:

- | | |
|----------------------------|--|
| Akiyanova F. Zh. | - Doctor of Geographical Sciences, Professor (Kazakhstan) |
| Seitkan A. | - PhD, (Kazakhstan) |
| Baysholanov S. S | - Candidate of Geographical Sciences, Associate professor (Kazakhstan) |
| Zayadan B. K. | - Doctor of Biological Sciences, Professor (Kazakhstan) |
| Salnikov V. G. | - Doctor of Geographical Sciences, Professor (Kazakhstan) |
| Mukanova A.S. | - PhD, (Kazakhstan) |
| Tasbolatuly N. | - PhD, (Kazakhstan) |
| Abdildayeva A. A. | - PhD, (Kazakhstan) |
| Chlachula J. | - Professor, Adam Mickiewicz University (Poland) |
| Redfern S.A.T. | - PhD, Professor, (Singapore) |
| Cheryomushkina V.A. | - Doctor of Biological Sciences, Professor (Russia) |
| Bazarnova N. G. | - Doctor Chemical Sciences, Professor (Russia) |
| Mohamed Othman | - Dr. Professor (Malaysia) |
| Sherzod Turaev | - Dr. Associate Professor (United Arab Emirates) |

Editorial address: 8, Kabanbay Batyr avenue, of.316, Nur-Sultan,
Kazakhstan, 010000
Tel.: (7172) 24-18-52 (ext. 316)
E-mail: natural-sciences@aiu.kz

International Science Reviews NST - 76153

International Science Reviews

Natural Sciences and Technologies series

Owner: Astana International University

Periodicity: quarterly

Circulation: 500 copies

CONTENT

Г.Қ.Тарина РЕИНТРОДУКЦИЯ ОСЕТРОВЫХ И МЕРЫ ПО ПРЕДОТВРАЩЕНИЮ РАСПРОСТРАНЕНИЯ ЧУЖЕРОДНЫХ ОСЕТРОВЫХ.....	5
Ш.Кәрім, М.Сыздыкова, А.Ерсин, А.Халық, А.Б.Карабалаева, Е.Н.Сағатбаев ОСНОВНЫЕ АСПЕКТЫ НОВОВВЕДЕНИЙ В ЭКОЛОГИЧЕСКОМ ЗАКОНОДАТЕЛЬСТВЕ РЕСПУБЛИКИ КАЗАХСТАН.....	10
N. Shaposhnikova, M. Johnson, A. Seitkan THE NURA RIVER CLEAN-UP FROM MERCURY	18
A.S.Kokinbek, A.B.Myrzagaliyeva, M.Zh.Zhumagul OBESITY IS A GLOBAL PROBLEM. KNOWLEDGE ABOUT OBESITY.....	26
С.Е.Базаров БҰҚТЫРМА СУ ҚОЙМАСЫНДАҒЫ КӨКСЕРКЕ (SANDER LUCIOPERCA LINNAEUS, 1758) МЕН ШОРТАННЫҢ (ESOX LUCIUS LINNAEUS, 1758) ЭКОЛОГИЯЛЫҚ-БИОЛОГИЯЛЫҚ СИПАТТАМАСЫ.....	29
А.М.Қызайымбек, Г.Ж.Тағанова ВОЗДЕЙСТВИЕ NO CODE И LOW CODE НА РАЗРАБОТЧИКОВ: ПРЕИМУЩЕСТВА И ОГРАНИЧЕНИЯ ДЛЯ БИЗНЕСА.....	36
А. Назырова, А. Муканова, М. Калдарова, Л. Кусепова МОДЕЛИРОВАНИЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ С ИСПОЛЬЗОВАНИЕМ СТОХАСТИЧЕСКОЙ МОДЕЛИ.....	45

ВОЗДЕЙСТВИЕ NO CODE И LOW CODE НА РАЗРАБОТЧИКОВ: ПРЕИМУЩЕСТВА И ОГРАНИЧЕНИЯ ДЛЯ БИЗНЕСА

Қызайымбек Айгерім Мақсатқызы

Таганова Гүлдана Жарқымбайқызы

Международный университет Астана, Астана, Қазақстан

ВТиПО, 2 курс

kyzainbek_aigerim@mail.ru

Аннотация. В современном мире, где технологические достижения меняют облик бизнеса и технологий, роль No Code и Low Code подходов в разработке программного обеспечения становится всё более существенной. Эти подходы позволяют создавать приложения и системы без необходимости глубокого понимания программирования, перераспределяя акценты в разработке. В этой статье анализируется, как No Code и Low Code влияют на роль разработчиков и динамику бизнеса, выявляются плюсы и минусы этих подходов, а также рассматривается их перспективное будущее.

Ключевые слова: No Code, Low Code, разработка программного обеспечения, роль разработчиков, влияние на бизнес, преимущества, ограничения, перспективы.

ВВЕДЕНИЕ

В современной эпохе, где центральное место занимают технологические инновации, эволюция методов разработки программного обеспечения продолжает активно развиваться. Одной из ключевых тенденций, определяющих этот процесс, стали подходы No Code и Low Code. В то время как традиционные методы программирования остаются важными, эти новаторские методологии предоставляют средства для создания приложений с минимальной или даже без вовлечения традиционных разработчиков.

No Code и Low Code - это понятия, которые за последние годы приобрели значительное значение в сфере технологической разработки. В основе этих подходов лежит идея упрощения и ускорения процесса создания программного обеспечения путем предоставления инструментов, которые позволяют даже не программистам создавать и настраивать приложения. No Code подразумевает использование готовых блоков и компонентов для создания приложений, не требуя написания кода вообще. Low Code предоставляет более гибкий подход, где можно использовать готовые компоненты, но также предоставляется возможность внедрения специфической логики с использованием небольшого количества кода.

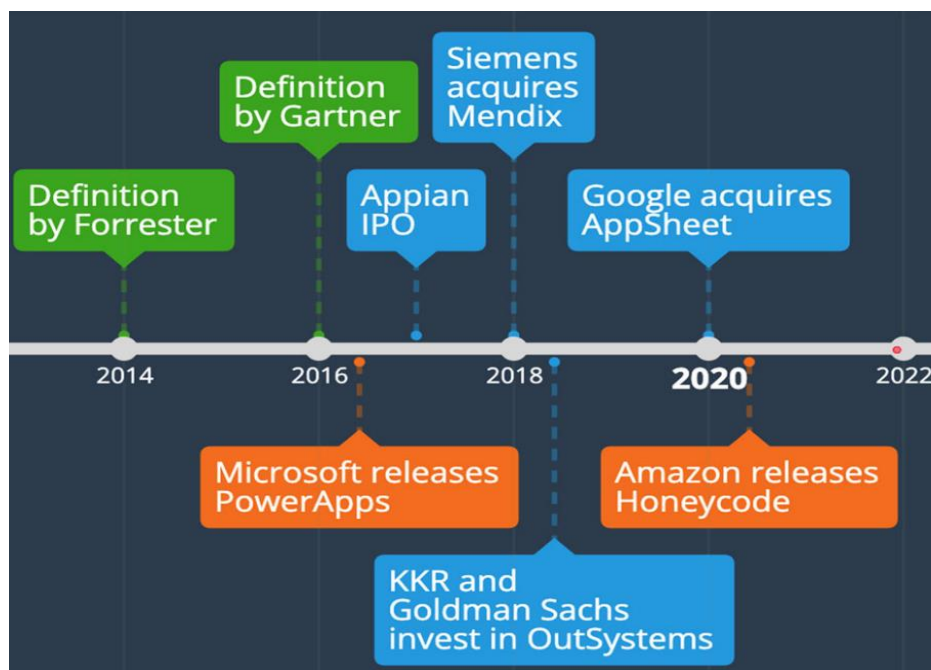


Рисунок-1. Основные события в истории low code

Эти подходы возникают в ответ на постоянно растущую потребность в быстрой разработке программных решений, которые могут адаптироваться к изменяющимся требованиям бизнеса. Актуальность исследования воздействия No Code и Low Code подходов на сферу разработки и бизнеса трудно переоценить. Они предоставляют компаниям средства для более быстрого внедрения инноваций, создания MVP для быстрого тестирования идеи на рынке, а также снижения затрат и зависимости от ограниченного числа высококвалифицированных разработчиков.

ИЗМЕНЕНИЕ РОЛИ РАЗРАБОТЧИКОВ

В современной динамичной области информационных технологий, преобразования в парадигме разработки программного обеспечения стали ключевым фактором, формирующим новый облик профессии разработчика. В контексте революционных подходов No Code и Low Code, традиционное понимание роли разработчика существенно эволюционировало, подчеркивая их стратегическое значение и их способность адаптироваться к новым технологическим парадигмам. Рассмотрим конкретные факты и примеры, которые демонстрируют это изменение.

Анализ изменения задач и обязанностей разработчиков

С появлением No Code и Low Code разработчики все чаще взаимодействуют с бизнес-аналитиками еще на более ранних этапах проекта. Они помогают перевести

требования бизнеса в технические решения и проектируют архитектуру приложений с учетом бизнес-целей.

Разработчики больше не тратят большую часть времени на написание каждой строки кода. Вместо этого они могут сосредоточиться на более высокоуровневых задачах, таких как выбор подходящих компонентов, оптимизация производительности и настройка бизнес-процессов.

Смещение акцентов

Разработчики теперь акцентируют свое внимание на создании гибкой и масштабируемой экосистемы приложений. Они интегрируют готовые компоненты и настраивают их взаимодействие, обеспечивая синергию между различными частями системы.

Роль разработчиков переходит к оптимизации и автоматизации бизнес-процессов. Они настраивают правила и логику работы системы, чтобы максимизировать эффективность использования приложения в рамках конкретного контекста.

Примеры навыков и знаний

Разработчики должны обладать глубоким пониманием архитектуры приложений, чтобы правильно организовать компоненты и обеспечить их взаимодействие.

Навыки работы с API и интеграциями становятся критически важными. Разработчики должны уметь связывать различные сервисы и приложения, чтобы создать совокупную функциональность.

Знание принципов оптимизации производительности и масштабируемости помогает создавать стабильные и эффективные приложения, даже если они разрабатываются с использованием No Code и Low Code.

Разработчики должны быть хорошими коммуникаторами, так как они тесно сотрудничают с разными членами команды - от бизнес-аналитиков до конечных пользователей.

ПЛЮСЫ И МИНУСЫ ДЛЯ РАЗРАБОТЧИКОВ:

Таблица-1. Плюсы и Минусы

ПЛЮСЫ	МИНУСЫ
Одним из ключевых плюсов No Code и Low Code подходов для разработчиков является возможность избавиться от рутины. Задачи, которые ранее требовали многократного кодирования, теперь могут быть автоматизированы благодаря готовым компонентам и интерфейсам.	Одним из ограничений No Code и Low Code подходов является сложность реализации высоко индивидуализированных и сложных решений. Пользовательские требования могут выходить за рамки стандартных компонентов, что может потребовать дополнительного кодирования или даже отказа от использования этих подходов.
No Code и Low Code позволяют быстро создавать прототипы приложений и быстро приступить к тестированию итеративных версий. Это особенно полезно на начальных этапах проектов, когда быстрый обратный отклик является ключевым фактором успеха.	В зависимости от выбранной No Code или Low Code платформы, могут возникнуть ограничения в интеграции с некоторыми сторонними сервисами или системами. Это может привести к ограниченности функциональности созданных приложений.
Освобождение разработчиков от рутинных задач позволяет им более эффективно вкладываться в сложные технические задачи и стратегическое планирование. Они могут сконцентрироваться на создании архитектуры, оптимизации производительности и настройке системы под уникальные потребности.	Использование No Code и Low Code подходов может привести к риску потери глубоких технических знаний и навыков программирования. Если разработчики полностью полагаются на готовые инструменты, это может снизить их способность понимать внутренние механизмы системы и принимать более сложные технические решения.

No Code и Low Code позволяют разработчикам расширить свои навыки в области интеграции различных компонентов и настройки сложных систем. Они становятся специалистами в области выбора и интеграции различных инструментов, что делает их ценными экспертами для бизнеса.	В случае сложных или высоконагруженных приложений, No Code и Low Code инструменты могут иметь ограниченные возможности оптимизации и тонкой настройки производительности. Разработчики могут столкнуться с ограничениями в оптимизации алгоритмов и структур данных.
Для стартапов и новых проектов создание минимально жизнеспособного продукта с использованием No Code и Low Code подходов может значительно ускорить время до выхода на рынок. Это позволяет быстро опробовать идею и получить обратную связь от пользователей.	При использовании конкретных No Code или Low Code платформ разработчики могут оказаться зависимыми от функциональности и ограничений этих инструментов. Это может ограничить их свободу выбора и гибкость при разработке.
No Code и Low Code инструменты облегчают внесение изменений и обновлений в приложения, поскольку большая часть логики может быть настроена без необходимости переписывания кода. Это упрощает обслуживание и адаптацию приложений к новым требованиям.	В случае интеграции с уже существующими сложными системами, No Code и Low Code подходы могут столкнуться с трудностями в настройке взаимодействия и обеспечении целостности данных между различными компонентами.

ВЛИЯНИЕ НА БИЗНЕС

Современный бизнес, несомненно, сталкивается с необходимостью быстрой адаптации к постоянно меняющейся технологической ландшафту. В этом контексте подходы No Code и Low Code представляют собой значительное влияние на способы, с которыми компании создают, развивают и масштабируют программные решения. Рассмотрим, как эти подходы воздействуют на бизнес.

Внедрение подходов No Code и Low Code открывает бизнесу новые возможности, позволяя ускорить процесс разработки и оптимизировать цикл

развертывания программных продуктов. Сокращение времени до запуска на рынок является одним из наиболее заметных плюсов. Ранее длительный процесс программирования заменяется готовыми компонентами и интуитивными интерфейсами, что позволяет значительно сократить временные рамки между идеей и воплощением на рынке.

Еще одним преимуществом является способность компаний быстро реагировать на изменения требований клиентов и динамичность рынка. Гибкость No Code и Low Code подходов позволяет бизнесу оперативно вносить изменения и адаптироваться к новым потребностям, что становится критически важным в быстро меняющейся среде бизнеса.

Следует также отметить, что использование этих подходов может снизить нагрузку на IT-отдел компании. Множество повседневных задач, которые ранее требовали активного участия разработчиков, теперь могут быть выполнены другими специалистами благодаря простоте и доступности инструментов. Таким образом, разработчики освобождаются от операционных задач и могут сфокусироваться на более сложных и стратегических аспектах.

Следует признать, что несмотря на множество преимуществ, No Code и Low Code подходы также сопряжены с некоторыми вызовами. Один из основных аспектов – это ограничение в развитии сложных продуктов. Готовые компоненты и шаблоны могут ограничить кастомизацию и возможность создания уникальных решений. Для компаний, работающих над проектами с уникальными требованиями, это может стать серьезным ограничением.

Также следует учитывать зависимость от конкретных платформ и поставщиков. Подходы No Code и Low Code могут в значительной степени опираться на функциональность выбранной платформы, что может ограничить гибкость бизнеса при переходе на другие технологические решения или усложнении интеграции с существующими системами.

Важным аспектом является также безопасность. Простота и скорость разработки с использованием этих подходов могут привести к недостаточному вниманию к аспектам безопасности и соблюдению стандартов. Ошибки в настройке безопасности или недостаточная проверка на соблюдение нормативов могут привести к уязвимостям и риску для бизнеса.

БУДУЩЕЕ NO CODE и LOW CODE

Постоянное развитие технологий привело к тому, что No Code и Low Code подходы стали неотъемлемой частью современной индустрии разработки ПО. Однако каково будущее этих методологий? Какие тенденции мы можем ожидать в долгосрочной перспективе и как они повлияют на разработчиков и бизнес? Давайте

попробуем предвидеть, как No Code и Low Code будут развиваться в будущем и как это отразится на нашем представлении о разработке программного обеспечения.

Вероятно, что в будущем No Code и Low Code подходы продолжат эволюционировать, становясь все более мощными и универсальными инструментами. Благодаря постоянному внедрению новых технологий и расширению возможностей, эти подходы будут способствовать созданию более сложных и инновационных приложений. Мы можем ожидать улучшенных инструментов для интеграции с различными системами, более гибких средств настройки, а также расширения функциональности для удовлетворения потребностей разнообразных отраслей.

В долгосрочной перспективе, разработчики будут продолжать менять свою роль под воздействием No Code и Low Code. Однако вместо угрозы для их профессиональной жизни, эти подходы будут служить инструментами расширения их способностей. Разработчики смогут сосредоточиться на высокоуровневых архитектурных решениях, создании инновационных стратегий и более тесном взаимодействии с бизнес-аналитиками и конечными пользователями. Это приведет к более продуктивному сотрудничеству и улучшенной способности воплощать идеи в жизнь.

Для бизнеса No Code и Low Code подходы будут продолжать приносить значительные выгоды. Внедрение приложений станет еще более быстрым и эффективным, что позволит бизнесам оперативно реагировать на изменения рынка и требований клиентов. Однако бизнесам также следует учитывать ограничения, такие как потребность в гибкости и кастомизации, чтобы успешно справляться с сложными и уникальными задачами.

КЕЙСЫ ИСПОЛЬЗОВАНИЯ

Применение No Code подхода породило ряд проектов и стартапов, которые успешно используют эту методологию для создания инновационных приложений, такие как: Bubble, Adalo, Glide.

Стартап-Bubble предоставляет инструмент для создания веб-приложений без использования кода. Благодаря платформе Bubble, даже лица без технической экспертизы могут разрабатывать сложные веб-сайты и приложения, от маркетплейсов до систем управления клиентами.

Adalo предоставляет возможность разрабатывать мобильные приложения без необходимости программирования. Благодаря графическому интерфейсу, пользователи могут создавать прототипы и разрабатывать мобильные приложения с разнообразной функциональностью.

Glide-эта платформа позволяет превращать данные из Google Sheets в интерактивные веб- и мобильные приложения. Проекты, созданные на Glide, позволяют быстро и просто разработать приложения на основе существующих данных.

Уже на данный момент компании Salesforce Lightning Platform и Unqork успешно используют no code/low code.

Salesforce, ведущий поставщик CRM-систем, предоставил клиентам Lightning Platform для разработки персонализированных приложений с минимальным кодированием. Это позволяет компаниям быстрее адаптироваться к изменениям и оперативно реагировать на новые требования рынка.

Платформа Unqork успешно применяет No Code методологию для создания сложных приложений и решений. Компании, включая Liberty Mutual, используют Unqork для разработки критически важных систем, таких как управление страховыми заявками.

ЗАКЛЮЧЕНИЕ

Революционные подходы No Code и Low Code несомненно изменили пейзаж разработки программного обеспечения, переосмыслив роль разработчиков и воздействуя на бизнес-процессы. Они предоставили множество преимуществ, которые нельзя недооценить. Вместе с тем, они также несут в себе ряд ограничений и вызовов. Итоговое влияние No Code и Low Code на разработчиков и бизнес зависит от внимательного анализа и адаптации подходов к конкретным задачам и стратегическим целям компании.

Применение No Code и Low Code подходов ускоряет процесс разработки, давая возможность быстро реагировать на изменения требований и быстрее выводить продукты на рынок. Более широкий круг специалистов может участвовать в создании приложений, что способствует демократизации разработки и появлению новых идей. Однако нельзя забывать о том, что No Code и Low Code не подходят для всех задач. Ограниченные возможности кастомизации и зависимость от платформ могут стать преградой для сложных и уникальных проектов, где требуется полное техническое владение.

Важно понимать, что выбор между No Code, Low Code и традиционным программированием не является жестким. Он зависит от сущности проекта, бизнес-стратегии и сроков. Возможно, оптимальным путем будет совмещение различных подходов в зависимости от контекста.

В заключение, эволюция в направлении No Code и Low Code подходов несомненно вносит существенный вклад в современную разработку программного обеспечения и бизнес-процессы. Осознанный выбор, учет преимуществ и

ограничений, а также адаптация подходов к потребностям компании позволят достичь максимальной эффективности и успешности в быстро меняющемся мире IT-инноваций.

СПИСОК ЛИТЕРАТУРЫ

1. Journal of Software Engineering Research and Development
2. Davide Di Ruscio, Dimitris Kolovos, Juan de Lara, Alfonso Pierantonio, Massimo Tisi, Manuel Wimmer "Low-code development and model-driven engineering: Two sides of the same coin?"
3. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering, 2nd edn. Morgan & Claypool Publishers, San Rafael (2017)
4. Hammond, J.: The Forrester Wave: Mobile Low-Code Development Platforms, Q1 2017. Forrester Research, Cambridge (2016)
5. Официальный сайт Microsoft Power Apps: <https://powerapps.microsoft.com/>
6. Официальный сайт Salesforce Lightning Platform: <https://www.salesforce.com/platform/low-code/>
7. Morrison, M. (2021). "No-Code Bible: How to Launch a No-Code Startup from Scratch and Scale to Profit."
8. Gartner. (2020). "Magic Quadrant for Enterprise Low-Code Application Platforms."
9. Forrester Research. (2021). "The Forrester New Wave™: Low-Code Platforms For Business Developers, Q4 2020."
10. Mendix. (2021). "2021 Mendix Low-Code Adoption Survey."
11. IEEE Software. (2020). "No-Code/Low-Code Development: A Systematic Mapping Study."
12. DevOps.com. (2021). "Why No-Code, Low-Code Is the Future of Application Development."