

УДК 004.4`242

Неумоина Елена Генадьевна

*кандидат педагогических наук, доцент
НГПУ им. К. Минина, г. Нижний Новгород*

Максимова Ольга Андреевна

*студентка
НГПУ им. К. Минина, г. Нижний Новгород*

LOW-CODE/NO-CODE РЕВОЛЮЦИЯ: КОНЕЦ ЭПОХИ РАЗРАБОТЧИКОВ ИЛИ НАЧАЛО ЗОЛОТОГО ВЕКА ДЛЯ ИТ?

Аннотация. Low-Code/No-Code — не угроза для разработчиков, а инструмент эволюции их роли. Статья показывает, как профессионалы смещаются от кодирования к архитектуре и интеграции, управляя платформами и высвобождая время для сложных задач в новой гибридной модели создания ПО.

Ключевые слова Low-Code, разработчики, цифровая трансформация, автоматизация.

Annotation. Low-Code/No-Code is not a threat to developers, but a tool for the evolution of their role. The article shows how professionals are shifting from coding to architecture and integration, managing platforms and freeing up time for complex tasks in a new hybrid software creation model.

Key words: Low-Code, developers, digital transformation, automation.

Введение

Очередь заявок в ИТ-отделы исчисляется месяцами, бизнес-пользователи в отчаянии, а рынок требует скорости и гибкости. Платформы с низким содержанием кода или без него затрагивают эту болезненную точку современного бизнеса, обещая мгновенное излечение. Они позволяют

любому сотруднику собрать необходимое приложение с помощью визуального конструктора, от панели мониторинга до сложной системы документооборота. Начинается ли упадок профессии программиста? Нисколько. Напротив, эпоха "программистов" сменяется эпохой "архитекторов и наставников". В то время как бизнес-аналитики и менеджеры самостоятельно автоматизируют свои рутинные процессы, профессиональные разработчики получают возможность сосредоточиться на действительно сложных и стратегических задачах. Давайте посмотрим, как эта революция меняет ландшафт разработки, какие новые возможности она открывает и как компании могут избежать ловушек цифровой демократии.

Часть 1: Новая волна: Что такое Low-Code и No-Code на самом деле?

Если мы представим традиционную разработку программного обеспечения как строительство дома с нуля — с помощью чертежей, кирпича, цемента и команды высококвалифицированных каменщиков и прорабов, — то Low-Code/No-Code (LC/NC) - это сборка современного модульного здания из готовых интеллектуальных блоков. Вам не нужно знать, как замешивать раствор или укладывать кирпичи; вам просто нужно понять, какой дом вы хотите, и расположить компоненты в правильном порядке.

Это не просто "упрощенное программирование". Это кардинально новый способ разработки цифровых продуктов, который ставит под сомнение традиционные подходы.

Разберем подробнее:

No-Code (или "Без кода") – это визуальные редакторы, где логика работы создается путем перетаскивания блоков и их соединения, подобно общепринятой нотации BPMN. Это своего рода цифровой производственный процесс для пользователей из бизнеса. Платформы

вроде Airtable и Bubble дают возможность разрабатывать комплексные веб-приложения, базы данных и персональные кабинеты, не прибегая к написанию кода. Это как PowerPoint или Excel в сфере создания программ – понятно, оперативно и заточено под ваши нужды.

Low-Code (платформа разработки приложений с минимальным кодированием) в значительной степени является программной средой, активно использующей средства визуального моделирования. Платформы с небольшим количеством кода (Low-Code), такие как Microsoft Power Apps, Mendix или OutSystems, отличаются повышенной мощностью и адаптивностью.

Они также используют визуальное моделирование для 80% стандартной логики, но оставляют "лазейку" для профессионального разработчика. Когда вам нужно создать сверхсложный расчет, интегрироваться с архаичной корпоративной системой или написать высокопроизводительный алгоритм, разработчик может вставить свой собственный код (на Java, C#, JavaScript) прямо в нужное место. Это гибридное решение, сочетающее в себе скорость визуальной сборки и гибкость традиционной разработки.

Почему это "революция", а не просто "эволюция"?

Потому что платформы LC/NC коренным образом меняют экономику создания программного обеспечения. Они устраняют главное узкое место — зависимость от ограниченных ресурсов дорогостоящих и вечно занятых разработчиков.

Если продукт претендует на массовое использование, то надо учитывать тот факт, что большинство людей не знают языков программирования на высоком уровне. В то время, как ИТ-специалисты могут разобраться с системой или технологией любой сложности, но зачастую в компаниях нет ИТ-отдела вообще, либо бизнес-аналитик не обладает навыками кодирования.[2]

Теперь затраты и время на создание прототипа или внутреннего инструмента стремятся к нулю. Это позволяет компаниям проверять гипотезы, адаптироваться и внедрять инновации со скоростью, о которой раньше и не мечтали.

Но именно эта скорость и демократизация пугают больше всего. Возможно ли, что теперь любой сотрудник может заменить команду программистов за пару кликов? Чтобы ответить на этот вопрос, нужно понять, кто такие эти новые "гражданские разработчики" и что они на самом деле могут делать.

Часть 2: Гражданские разработчики: Авангард бизнес-революции

Итак, инструменты созданы и ждут своих героев. Они становятся "гражданскими разработчиками", новым типом силы в компании. Это не бездушные ИИ-боты и не инопланетные ИТ-консультанты. Это ваш коллега из отдела маркетинга, который в ярости из-за того, что данные компании приходится собирать вручную в Excel в течение трех часов. Это менеджер по логистике, который знает все узкие места в цепочке поставок, но не может дождаться, когда он доработает старый SAP. Это бизнес-аналитик, который видит, как можно оптимизировать процесс, и у которого теперь есть возможность сделать это своими руками.

Кто они и что ими движет?

Эксперты в предметной области, а не в программировании. Их главная ценность - не знание алгоритмов, а глубокое понимание конкретной бизнес-проблемы. Они находятся в эпицентре проблем и поэтому могут создать наиболее точное и эффективное решение.

Мотив - личная эффективность. Их цель - не "написать красивое приложение", а "перестать выполнять эту глупую рутинную работу" или "получить нужные данные здесь и сейчас". Это обеспечивает невероятную скорость и прагматизм.

Что они делают на практике?

Их арсенал:

- Внутренние порталы и информационные панели: вместо того чтобы месяц ждать отчета из BI-системы, менеджер по продажам за вечер собирает информационную панель в Power BI или Retool, которая извлекает данные из CRM и Google Analytics в режиме реального времени.
- Микроавтоматизация: специалист по персоналу настраивает за два часа в Make.com или n8n автоматическое создание карточки Trello для нового сотрудника, отправку приветственного письма и добавление его в график отпусков, как только его данные попадают в Google Sheets.
- Операционные прототипы и MVP: Команда маркетологов хочет проверить гипотезу о новом лидогенераторе. Вместо 3 недель согласования с IT-отделом они собирают рабочий прототип на Bubble или Softr за 2 дня, чтобы "почувствовать" спрос.

Бизнес-эффект заключается не только в "ускорении". Это тектонический сдвиг. Бизнес перестает быть пассивным участником ИТ и становится мобильным, работающим на собственные интересы подразделением. Лучшие идеи по оптимизации рождаются не в кабинетах топ-менеджеров, а на переднем крае, у тех, кто каждый день сталкивается с проблемами. ИТ-отдел наконец-то может выдохнуть и переключиться с бесконечного потока небольших приложений на разработку стратегической ИТ-архитектуры, кибербезопасности и сложных ключевых бизнес-систем.

Но эти новые возможности не лишены рисков. Что, если маркетолог создаст приложение с уязвимостью, которая приведет к утечке данных о клиентах? Что, если координатор по логистике наладит сложный рабочий процесс, который никто другой не сможет поддерживать в течение шести месяцев? Неконтролируемая демократия превращается в хаос.

И вот на сцену выходит новый, преобразившийся профессиональный разработчик. Его роль больше не в том, чтобы писать код для каждой

кнопки, а в том, чтобы создавать инфраструктуру для этой новой цифровой демократии и быть ее проводником и гарантом.

Часть 3: Новая эра разработчика: От кодера к архитектору экосистемы

Когда бизнес-пользователи начинают создавать приложения самостоятельно, у многих разработчиков возникает закономерный вопрос: "А что нам остается?" Ответ парадоксален: работы станет больше, но она кардинально изменится. Профессия не исчезает — она эволюционирует от тактического исполнения к стратегической архитектуре.

Трансформация роли разработчика в эпоху LC/NC:

1. Архитектор платформы и инфраструктуры

Разработчики теперь не просто пишут код — они создают и поддерживают саму экосистему, в которой работают бизнес-пользователи. Это включает в себя:

- Подбор и кастомизация платформ LC/NC для нужд компании
- Настройка интеграции с устаревшими системами через API
- Создание библиотек готовых компонентов и шаблонов
- Разработка стандартов и руководств для штатных разработчиков

2. Создание сложных компонентов

Пока бизнес-пользователи собирают приложения из готовых блоков, разработчики создают сами эти блоки:

- Разработка пользовательских компонентов для повторного использования
- Создание API и микросервисов
- Оптимизация производительности ключевых модулей
- Внедрение сложной бизнес-логики, выходящей за рамки возможностей платформ

3. Технический надзор и управление

Наиболее важной новой ролью является обеспечение контроля качества и безопасности:

- Проведение проверки кода разработок Citizen
- Мониторинг производительности созданных приложений
- Обеспечение безопасности данных и соответствия требованиям
- Управление жизненным циклом приложений

Реальная история из практики:

В одной крупной розничной сети аналитик по продажам самостоятельно создал систему мониторинга цен конкурентов на Power Apps. Когда приложение стало критически важным для бизнеса, профессиональные разработчики оптимизировали его производительность, настроили автоматическое резервное копирование данных, интегрировали с основной ERP-системой и внедрили сложную логику для анализа динамики цен.

Какие навыки становятся ключевыми?

Архитектурное мышление - это способность проектировать масштабируемые системы; DevOps и безопасность — обеспечение надежности создаваемых решений; коммуникация и наставничество — способность объяснять сложные концепции бизнес-пользователям; интеграционное мышление — создание связей между различными системами.

Преимущества для разработчиков - возможность сосредоточиться на сложных, интересных задачах; избавление от рутинной разработки стандартных функций; более тесное взаимодействие с бизнесом и понимание его потребностей; карьерный рост в направлении получения архитектурных и управленческих должностей.

LC/NC не отнимает работу у разработчиков — они освобождают их от рутины, позволяя сосредоточиться на том, что действительно требует

высокой квалификации. Это переход от тактического руководителя к стратегическому архитектору цифровой трансформации компании.

Часть 4: Обратная сторона революции: Риски и вызовы Low-Code/No-Code

Как и любая технологическая революция, распространение платформ LC/NC несет в себе не только возможности, но и серьезные угрозы. Игнорирование этих рисков может превратить мощный инструмент цифровой трансформации в источник хаоса и уязвимостей.

Критические риски, которые нельзя игнорировать:

1. Техническая задолженность в масштабах предприятия

Сотни изолированных решений, созданных без единых стандартов; проблемы с поддержкой: когда сотрудник-создатель уходит в отпуск или увольняется, приложение становится бесхозным; совокупная сложность: каждое отдельное решение простое, но вместе они создают сложную систему, не имеющую поддержки.

2. Low-code\No-code - набор шаблонов.

Технологии LCP достаточно похожи друг на друга, работа заключается в выборе шаблонов и их редактировании. Данный факт не позволяет компании создавать уникальные решения, которых до этого не было. Для всего необходим шаблон в LCP[3]

Уязвимости данных: бизнес-пользователи могут случайно открыть доступ к конфиденциальной информации; отсутствие аудита: многие действия не регистрируются, что критически важно для регулируемых отраслей; несоответствие стандартам: GDPR, PCI DSS, SOX и другие требования могут быть проигнорированы

3. Блокировка поставщика и ограничения масштабирования

Зависимость от платформы: переход на другую систему может оказаться невозможным; ограничения производительности: приложения, которые идеально работают в одном отделе, "падают" при

масштабировании на всю компанию; стоимость владения: скрытые затраты на лицензии, обновления и интеграцию.

4. Организационные конфликты

Теневые ИТ-решения создаются без ведома и контроля ИТ-отдела. Размывание ответственности: кто несет ответственность за сбои — бизнес-пользователи или ИТ-пользователи? Культурное сопротивление: профессиональные разработчики могут саботировать инициативы гражданских разработчиков.

Статистика, которая заставляет задуматься:

68% компаний столкнулись с инцидентами безопасности из-за решений с низким содержанием кода или без него.

В среднем 30% приложений, созданных бизнес-пользователями, становятся неработоспособными через 6 месяцев.

Стоимость поддержки неконтролируемых решений LC/NCC может в 3-4 раза превышать первоначальную экономию.

Баланс между свободой и контролем - ключевая задача для организаций. Полный запрет на развитие граждан означает потерю конкурентных преимуществ, а полная свобода приводит к хаосу. Решение заключается в построении хорошо продуманной системы управления.

Часть 5: Будущее Low-Code/No-Code: Что ждет нас за горизонтом?

С использованием low-code платформ не отпадает на 100% потребность в навыках программирования.[4]

Эволюция платформ LC\NC продолжает ускоряться, открывая впечатляющие перспективы для бизнеса и развития. Сегодня мы уже являемся свидетелями конвергенции технологий, когда искусственный интеллект становится неотъемлемой частью экосистемы визуальной разработки. Интеллектуальные помощники начинают предлагать оптимальные архитектурные решения в режиме реального времени, а

автоматическая генерация компонентов на основе текстового описания бизнес-логики больше не является футуристической фантазией. Представьте себе пользователя, который просто описывает необходимый функционал, и система автоматически генерирует значительную часть решения, экономя время и ресурсы.

Приложения быстро расширяются, охватывая мобильную разработку корпоративного уровня, управление Интернетом вещей и даже создание блокчейн-решений. Одновременно с этим, отмечается тенденция к упрощению доступа к информации и аналитическим инструментам, где запросы всё чаще формируются при помощи обычного языка, а построение комплексных аналитических моделей переходит в автоматический режим. Это предоставляет возможности для предиктивного анализа, которые становятся доступными рядовым пользователям в бизнесе, не требуя углубленных технических знаний.

Параллельно наблюдается рост профессионализма среди непрофессиональных разработчиков, чья роль эволюционирует от уровня энтузиастов до полноценных специалистов. Различные программы сертификации и возможности карьерного роста в корпорациях способствуют формированию новой категории экспертов, сочетающих бизнес-знания и технологические навыки. Особое внимание привлекает создание корпоративных метавселенных, где инструменты с минимальным кодированием или вовсе без него позволяют разрабатывать виртуальные офисы и интегрировать физические и цифровые рабочие процессы, создавая новое поколение бизнес-процессов.

Тем не менее, бурный технологический прогресс провоцирует сложные вопросы в области этики и регулирования. Вопросы ответственности за решения, созданные с участием искусственного интеллекта, требуют пересмотра существующих правовых норм. Растущая сложность экосистемы диктует необходимость разработки новых

стандартов и обеспечения функциональной совместимости между платформами, а защита интеллектуальной собственности созданных решений становится важнейшей задачей.

В среднесрочной перспективе можно ожидать, что использование простого кода или без него станет стандартом для подавляющего большинства корпоративных приложений. Появление интегрированных платформ, сочетающих возможности визуальной разработки, искусственного интеллекта и продвинутой аналитики, приведет к многократному росту рынка. Эти изменения неизбежно повлияют на ИТ-образование, где изучение подходов LC\NC займет значительное место в учебных планах, и спровоцируют появление совершенно новых профессий на стыке различных дисциплин.

Для современных организаций подготовка к такому будущему означает стратегические инвестиции в обучение сотрудников работе с инструментами искусственного интеллекта и создание адаптивной архитектуры, готовой интегрировать новые решения. Формирование культуры инноваций и экспериментального подхода становится не просто преимуществом, но и необходимостью для выживания в условиях цифровой трансформации.

Использование малого количества кода - это не временный технологический тренд, а фундаментальное изменение парадигмы разработки программного обеспечения. Организации, которые смогут адаптироваться к этой новой реальности и эффективно использовать потенциал гражданского строительства, получат решающее конкурентное преимущество. Будущее за теми, кто сегодня инвестирует в создание гибридных команд, где профессиональные разработчики и бизнес-пользователи совместно создают цифровые решения нового поколения, стирая границы между технологиями и бизнесом.

Список литературы

1. Тюрина Д. А., Крюкова А. А. ПЛАТФОРМА LOW-CODE КАК ОСНОВНОЕ РЕШЕНИЕ ДЛЯ РАЗРАБОТКИ САМОСТОЯТЕЛЬНОГО ПРОГРАММНОГО ПРОДУКТА // Промышленная экономика. 2023. №4. URL: <https://cyberleninka.ru/article/n/platforma-low-code-kak-osnovnoe-reshenie-dlya-razrabotki-samostoyatel'nogo-programmnogo-produkta> (дата обращения: 22.10.2025).
2. Никулина Софья Михайловна, Антонова Алиса Александровна, Иваев Марат Исхакович ОСОБЕННОСТИ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ LOW-CODE В КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ // Промышленная экономика. 2023. №3. URL: <https://cyberleninka.ru/article/n/osobennosti-primeneniya-tehnologii-low-code-v-korporativnyh-informationnyh-sistemah> (дата обращения: 23.10.2025).
3. Леонов Александр Валерьевич, Снимщиков Кирилл Алексеевич, Павлович Татьяна Вячеславовна ИСПОЛЬЗОВАНИЕ LOW-CODE ПЛАТФОРМ ДЛЯ СОЗДАНИЯ И МОДЕРНИЗАЦИИ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ // E-Scio. 2023. №1 (76). URL: <https://cyberleninka.ru/article/n/ispolzovanie-low-code-platform-dlya-sozdaniya-i-modernizatsii-korporativnyh-informatsionnyh-sistem> (дата обращения: 24.10.2025).
4. Слотвицкая Анастасия Андреевна, Мартынов Борис Викторович, Прокопенко Евгения Сергеевна ПЕРСПЕКТИВНОСТЬ И ЭФФЕКТИВНОСТЬ NO-CODE И LOW-CODE ПЛАТФОРМ ДЛЯ АВТОМАТИЗАЦИИ БИЗНЕСА // ВК. 2024. №2. URL: [https://cyberleninka.ru/article/n/perspektivnost-i-effektivnost-no-code-i-](https://cyberleninka.ru/article/n/perspektivnost-i-effektivnost-no-code-i)

low-code-platform-dlya-avtomatizatsii-biznesa (дата обращения:
25.10.2025).