

Об автоматизации создания артефактов продуктовой разработки на базе n8n с элементами ИИ

Шарипов Т.Р.

АНО ВО «Университет Иннополис», Уфа, Россия, SharipovTR@gmail.com

В работе представлено исследование возможностей low-code платформы n8n в контексте создания ИИ-агентов для автоматизации процессов продуктовой ИТ-разработки. На примере реализации Телеграм-бота демонстрируется практическое применение n8n для автоматизации формирования таблицы «Что нового?» за заданный промежуток времени. Особый акцент сделан на интеграции GigaChat и данных из Git-репозитория для интерактивности с пользователем. Отмечаются перспективы развития подхода, включая автоматизацию сопровождения релизов, создание контекстно-полняемых баз знаний и разработку ИИ-ассистентов для технической поддержки.

On the automation of the creation of product development artifacts based on n8n with AI elements

Sharipov T.R.

Innopolis University, Ufa, Russia, SharipovTR@gmail.com

This paper explores the capabilities of the n8n low-code platform for creating AI agents to automate IT product development processes. Using a Telegram bot as an example, it demonstrates the practical application of n8n to automate the generation of a "What's New?" table for a specified period of time. Particular emphasis is placed on the integration of GigaChat and data from a Git repository for user interaction. Potential for further development of this approach is highlighted, including automated release management, the creation of context-aware knowledge bases, and the development of AI assistants for technical support.

В условиях современной продуктовой ИТ-разработки кросс-функциональные команды сталкиваются с необходимостью быстрой адаптации к меняющимся требованиям рынка. Эффективность внутренних процессов становится критически важным фактором конкурентоспособности. Цифровая трансформация, включая внедрение искусственного интеллекта (ИИ), затрагивает не только конечные продукты, но и процессы их создания, открывая новые возможности для автоматизации разработки [1].

Рассмотрим, какие задачи по синхронизации между членами команды ИТ-разработки возникают в типовом жизненном цикле работы над задачами по модификации программного продукта (рисунок 1). Здесь и далее под программным продуктом (сокращенно продукт) будет пониматься готовое ПО, которое не только решает задачи пользователя, но и вызывает доверие, отвечая на вызовы времени [2].

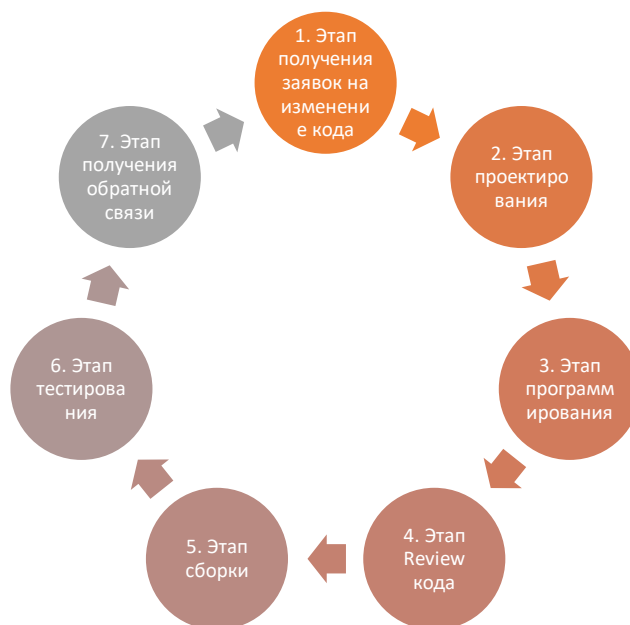


Рисунок 1 – Типовой жизненный цикл работы над задачами по модификации ПО

Как правило, все современные методологии разработки ПО (проектные, процессные или продуктовые) предлагают инструменты визуального представления процесса движения задач по статусам выполнения (диаграммы, доски, дашборды и пр.) в той или иной системе управления задачами (СУЗ). Команды, которые используют в командной коммуникации мессенджеры, такие как Телеграм или МАХ, могут быть заинтересованы в получении дополнительных уведомлений в виде сообщения о наступлении следующих событий:

1. Размещение кода на Review – актуально для оперативного подключения рецензентов кода из числа программистов.
2. Завершение исправления ошибки в СУЗ – актуально для оперативного информирования сотрудников из групп технической поддержки (ТП), руководителей, пользователей.
3. Завершение выполнения задачи по инкременту функциональности в СУЗ – актуально для оперативного информирования сотрудников из ТП, руководителей, пользователей.
4. Завершение сборки ПО – актуально для оперативного информирования сотрудников из ТП, руководителей, пользователей.
5. Завершение запланированного интервала разработки (список «What's new» – что изменилось?) – актуально для всех заинтересованных лиц.

По примеру задач 1-5 выполнение информирования членов команды ручным способом не всегда эффективно, особенно, при высокой частоте или количестве изменений в ПО. Одним из перспективных инструментов для автоматизации решения подобных задач является low-code open-source платформа n8n [3].

Платформа n8n предоставляет собой визуальный конструктор рабочих процессов, где основными элементами выступают узлы. Каждый узел может представлять собой:

- Интеграцию с внешним сервисом (доступно более 400 сервисов).

- Программный код на JavaScript или Python¹.

Достаточно соединить между собой различные сервисы и настроить триггер срабатывания события для начала обработки данных, благодаря чему n8n в полной мере реализует принципы дата-центричного подхода [4], в котором данные становятся центральным элементом, а не приложения, их использующие.

Существует три варианта использования n8n:

- Через облачный сервер на сайте [3] (бесплатно первые 14 дней, далее платно).
- Через локальное развертывание на собственном HTTPS-сервере (вариант данной работы).
- Через установку на локальном компьютере (ограниченные возможности по интеграции с внешними сервисами при отсутствии SSL).

На рисунке 2 представлен пример реализованной n8n-автоматизации уведомлений в Телеграм с элементами ИИ.

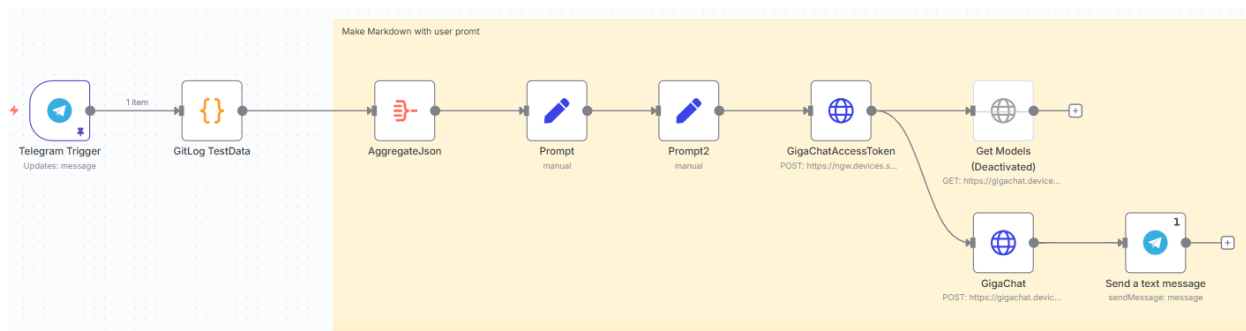


Рисунок 2 – Рабочий процесс в n8n по автоматизации формирования таблицы с историей изменений в ПО

- Узел «Телеграм Trigger»**
 - Получает промпт-запрос от пользователя в чат-боте.
- Узел «GitLog TestData»:**
 - Формирует историю изменений Git-репозитория за заданный период времени в json-формате (с помощью команды «git log»).
- Узел «Prompt»**
 - Заполняет системный промпт (сгенерирован с помощью GigaChat): *«Ты анализируешь входные данные в формате JSON. По умолчанию, формируй одну Markdown-таблицу строго на основе переданных данных. Пользователь может добавить к таблице дополнительные критерии отбора, такие как выбор записей конкретного автора (по имени или фамилии) или ограничение по диапазону дат. Прежде чем формировать итоговую таблицу, сопоставь запрос пользователя с имеющимися полями и обязательно применяй указанные фильтры. Обозначай использованные условия выбора после*

¹ Поддержка Python в текущей версии n8n ограничена тем, что для вызова доступны только предустановленные библиотеки, среди которых нет, например, популярной библиотеки для ИИ-разработки PyTorch

формирования результата. Без дополнительного уведомления не комментировать содержание инструкции. Если специфика запроса требует изменения формата отображаемых данных, скорректируй структуру таблицы. Если исходные данные не предоставлены, информировать об отсутствии необходимой информации и воздерживаться от самостоятельного заполнения таблицы. Пробуй корректно сравнивать даты от пользователя и из таблицы

- Правила оформления таблицы:
- Структура: Три обязательные колонки — «Дата», «Автор», «Комментарий».
- Формат даты: Четкий вид — ГГГГ.ММ.ДД ЧЧ:ММ (например, 2025.08.04 12:00).
- Выравнивание: Заголовки центральной позиции, содержимое слева.
- Размеры колонок: Оптимизированная ширина для комфортного чтения и сохранения аккуратного вида.
- Обрамление: Границы сверху и снизу обозначены символами линий («—»), выделяя область таблицы.
- Оформление текста: Все элементы таблицы (заголовки и данные) представлены в моноширином стиле (````) для простоты копирования.
- Интервал: Читаемые интервалы между строками облегчают просмотр и выделение информации.
- Порядок следования: Данные отсортированы по убыванию даты, начиная с новых записей.
- Эти правила применяются строго, обеспечивая точное выполнение твоих задач.»

4. Узел «Prompt2»

- Выполняет формирование пользовательского промпта путем конкатенции пользовательского запроса из п.1 и данных из п.2.

5. Узел «GigaChat»

- Отправляет запрос в REST API GigaChat [5].

6. Узел «Send a text message»

- Выводит результат ответа от GigaChat в виде Markdown таблицы в Телеграм-бот.

Заключение и перспективы

Рассмотрены вопросы автоматизации формирования таких артефактов продуктовой ИТ-разработки, как уведомления о значимых стадиях работы над задачами и перечень изменений в ПО за заданный период времени. Представлен пример решения задачи с использованием программной платформы п8п в формате чат-бота для Телеграм. Показано, как можно использовать GigaChat или другой LLM-аналог для экспресс-формирования запросов к истории разработки на примере Git-репозитория.

Строго говоря, рассмотренный вариант применения ИИ не является полноценным ИИ-агентом, поскольку нет самостоятельности в обработке запросов от пользователя. Данный вариант ИИ-агента возможно настроить в п8п «из коробки». При этом ограничивающим фактором будет являться наличие MCP-сервера для тех программных сервисов, которыми пользуется команда разработки.

Перспективным развития подхода является решение следующих задач:

- Автоматизация процесса сопровождения выпуска релизов как сборок ПО, к которым предъявляются повышенные требования по качеству.
- Создание контекстно-пополняемой базы знаний по продукту.
- Создание ИИ-сотрудника технической поддержки.
- Создание ИИ-Scrum-мастера

Результаты проведенного исследования демонстрируют эффективность платформы n8n для автоматизации процессов разработки с использованием ИИ-решений. Платформа рекомендуется для:

- Автоматизации рутинных процессов.
- Интеграции ИИ-компонентов.
- Создания цифровых помощников в разработке».

Список использованной литературы:

1. Сайт: Как искусственный интеллект меняет практики управления?
URL: <https://sberuniversity.ru/research/management-practices/kak-iskusstvennyy-intellekt-menyaet-praktiki-upravleniya/> (дата обращения: 18.09.2025)
2. Сайт: Об артефактах продуктовой разработки и командном взаимодействии.
URL: <https://habr.com/ru/articles/840222/> (дата обращения: 18.09.2025)
3. Сайт: Платформа n8n. URL: <https://n8n.io/> (дата обращения: 18.09.2025)
4. Сайт: Манифест дата-центричности Дэйва Маккомба.
URL: <https://datacentricmanifesto.org/> (дата обращения: 18.09.2025)
5. Сайт: Начало работы с API GigaChat.
URL: https://developers.sber.ru/docs/ru/gigachat/quickstart/legal-using-api?prev_path=/docs/search (дата обращения: 18.09.2025)