

AI-Driven Development: использование ИИ-агентов в разработке программного обеспечения

Уважаемые коллеги, добрый день.

Сегодня мы поговорим о теме, которая стремительно меняет современную разработку — это AI-Driven Development, то есть создание программного обеспечения с активным участием автономных ИИ-агентов.

Искусственный интеллект уже давно помогает программистам: подсказывает код, исправляет ошибки. Но последние достижения привели к появлению нового феномена — почти независимых ИИ-разработчиков, которые действуют не только по команде человека, но и самостоятельно анализируют проект, принимают мелкие решения, взаимодействуют друг с другом и выполняют реальные задачи в кодовой базе.

Сегодня мы разберём, что делает этот подход возможным, какие роли выполняют ИИ-агенты, какова их реальная автономность, какие преимущества и риски несёт новая парадигма, и каким будет будущее профессии разработчика.

Введение:

Сегодня под ИИ-агентами понимают не просто инструменты, которые реагируют на команды человека. Это почти автономные сущности, работающие как виртуальные удалённые разработчики. Они способны самостоятельно анализировать проект, принимать небольшие решения и выполнять задачи без постоянного потока промптов.

Современный ИИ-агент может:

- отслеживать изменения в репозитории, понимать состояние проекта и контекст задач;
- инициировать работу сам — например, улучшить структуру модуля или предложить тесты, если их не хватает;
- принимать локальные архитектурные и кодовые решения в рамках заранее заданных правил команды;
- обращаться к другим агентам или к разработчику только тогда, когда ему действительно нужно уточнение;
- запускать тесты, анализировать их результаты и вносить исправления без ручного вмешательства.

Иначе говоря, ИИ-агент — это не инструмент, который ждёт команду «сгенерируй мне вот это». Это постоянно работающий участник разработки, который сам проявляет инициативу и закрывает часть задач, ранее выполнявшихся младшими или средними разработчиками.

Главная идея — перенести рутинные и предсказуемые задачи на ИИ, в то время как человек сосредотачивается на архитектуре, стратегических решениях, рисках и бизнес-логике.

Организация работы с ИИ-агентами

Современные команды разработки могут существенно повысить эффективность, интегрируя ИИ-агентов в процессы создания и поддержки программного обеспечения.

Ниже описана структура такой работы, которая сочетает автоматизацию и человеческий контроль.

1. Роли ИИ-агентов и человека

ИИ-агенты берут на себя:

- Генерацию и сопровождение кода — создание функций, модулей, написание документации.
- Тестирование — юнит-тесты, интеграционные тесты, формирование баг-репортов.
- DevOps задачи — настройка CI/CD, контейнеризация, мониторинг инфраструктуры.

Человек сосредотачивается на:

- Стратегических решениях и архитектуре проектов.
- Ревью критически важных изменений.
- Контроле безопасности и управлении рисками.
- Принятии нестандартных и творческих решений, которые пока трудно делегировать ИИ.

2. Разделение задач и процессов

- Рутинные и повторяющиеся задачи полностью или частично автоматизируются агентами.
- Креативные и нестандартные задачи выполняются человеком.

Каждому ИИ-агенту задаются чёткие правила работы и ограниченный контекст проекта, чтобы он мог действовать автономно и не выходить за рамки своей зоны ответственности.

3. Автономная работа и коммуникация

ИИ-агенты могут:

- Инициировать задачи без прямой команды человека.
- Сотрудничать с другими агентами для согласования изменений.
- Обращаться к разработчику только при необходимости уточнения или при критических ситуациях.

Такой подход создаёт «умную» сеть, где агенты работают самостоятельно, но остаются под надзором человека.

4. Интеграция с CI/CD и системами управления проектом

- Агенты подключаются к репозиториям, трекерам задач и системам тестирования.
- Автоматически запускаются пайплайны, проводятся проверки, деплой и обновление документации.

Это ускоряет процессы и уменьшает количество человеческих ошибок при рутинных операциях.

5. Контроль качества и ревью

- Все критические изменения проходят ревью человеком.
- Используются автоматические проверки качества кода и тестов, чтобы минимизировать риск ошибок от ИИ.

Таким образом сохраняется баланс между скоростью разработки и надёжностью продукта.

6. Итеративное улучшение методологии

- На основе анализа работы агентов и метрик проекта правила и модели их поведения регулярно обновляются.
- Команда оценивает эффективность агентов, корректирует их зоны ответственности и адаптирует подход к новым задачам.

Это позволяет со временем создавать всё более автономные и «умные» системы, где ИИ-агенты становятся полноценными участниками процесса разработки.

ИИ-агенты в разработке: типы и функции

Чтобы лучше понять масштабы изменений, рассмотрим основные виды ИИ-агентов, которые используются в разработке.

Агенты генерации и сопровождения кода

Они умеют:

- писать функции, классы и модули по описанию;
- подбирать оптимальные алгоритмы;
- предлагать улучшения кода;
- объяснять чужой код;
- автоматически исправлять баги и формировать pull-requests.

Такие агенты значительно сокращают время разработки, особенно при работе с рутинными задачами.

Агенты тестирования

ИИ особенно полезен в тестировании:

- автоматически генерирует юнит-тесты и интеграционные тесты;
- проверяет типичные сценарии использования;
- предлагает баг-репорты;
- анализирует причины падений;
- формирует рекомендации по улучшению покрытия тестами.

DevOps-агенты

Эти агенты работают с инфраструктурой:

- создают CI/CD пайплайны;
- выбирают оптимальные параметры для сборки;
- анализируют логи и выявляют аномалии;
- управляют контейнерами и облачными сервисами;
- прогнозируют нагрузки и возможные сбои;
- генерируют конфигурации Terraform, Helm, Dockerfile и др.

Агенты документации

- Эти агенты автоматизируют создание и поддержку технической документации:
- создают README, внутренние инструкции и описания API;
- автоматически обновляют документацию при изменениях в коде;
- генерируют changelog и release notes;
- создают поясняющие схемы, диаграммы и архитектурные описания;
- формируют пользовательские и разработческие руководства.

Агенты анализа и ревью кода

Отдельная категория агентов, которые помогают контролировать качество:

- проводят code review;
- анализируют архитектуру и выявляют анти-паттерны;
- ищут потенциальные уязвимости и небезопасные конструкции;
- предлагают рефакторинг и оптимизацию;
- проверяют корректность ИИ-сгенерированного кода.

Агенты продуктовой и бизнес-аналитики

Их используют в крупных командах для автоматизации анализа требований:

- преобразуют требования в технические задачи;
- анализируют метрики использования продукта;
- прогнозируют влияние изменений на roadmap.

Лучшие модели для работы в агентном режиме

- Anthropic — Claude

Подходит для глубокого анализа кода и архитектуры, ревью и тестирования.

- Google — Gemini

Хорош для DevOps, многоагентных сценариев и генерации документации.

- OpenAI — o1/o3/o1-mini

Эффективны для планирования и анализа сложных процессов.

- GitHub — Copilot Agent

Генерирует код, pull-requests, тесты и обновляет документацию.

Преимущества AI-Driven Development

Автономность и ускорение разработки

- Агенты могут самостоятельно писать код, тесты и документацию, планировать задачи и выполнять рутинные операции.
- Снижение времени на выполнение повторяющихся задач до 30–50% по данным кейсов стартапов и корпоративных внедрений.

Повышение продуктивности команды

- Разработчики могут сосредоточиться на архитектуре, дизайне и сложных задачах, вместо рутины.

- Крупные компании отмечают рост эффективности разработчиков в 2 раза и уменьшение ошибок.

Быстрая адаптация к новым технологиям

- Агенты способны обучаться новым библиотекам, фреймворкам и API быстрее, чем команда осваивает их вручную.
- Это позволяет компании быстрее внедрять современные решения и интеграции.

Снижение дефектов, связанных с человеческим фактором

- Рутинные операции, например генерация шаблонного кода, тестов или скриптов деплоя, выполняются без усталости или невнимательности.
- Снижает количество багов, связанных с человеческим фактором.

Непрерывное тестирование и контроль качества

- Агенты могут автоматически запускать юнит-тесты, интеграционные тесты, проверять производительность и безопасность.
- Позволяет быстрее обнаруживать проблемы и ускоряет выпуск обновлений.

Снижение стоимости проекта

- Стоимость разработки уменьшается за счет автоматизации целых этапов. Особенно это важно для стартапов.

Недостатки AI Driven Development

Генерация логически неверного кода

- ИИ агенты иногда создают код, который выглядит аккуратно и убедительно, но при детальной проверке оказывается логически некорректным.
- Требуется постоянного контроля со стороны разработчиков и тестирования.

Вопросы безопасности

- Автоматически сгенерированный код может содержать уязвимости или небезопасные решения, особенно при работе с критическими системами.
- Необходим дополнительный аудит безопасности и проверка корпоративных стандартов.

Авторство и юридическая ответственность

- Код, созданный ИИ, поднимает вопросы о праве собственности и ответственности за ошибки.
- На данный момент нет универсальных юридических норм, регулирующих использование кода, сгенерированного ИИ.

Соппротивление внутри команд

- Некоторые разработчики опасаются, что ИИ заменит их.
- На практике ИИ автоматизирует рутинные задачи, позволяя людям сосредоточиться на сложных и творческих аспектах работы.

Ограниченность возможностей

- ИИ может не справиться с полностью новыми или нестандартными задачами, где нет готовых паттернов или данных.
- Для таких задач требуется глубокий анализ, исследование и творческий подход человека.

Практические кейсы использования ИИ агентов

1. Altan — стартап с командой ИИ агентов

Испанский стартап Altan построил платформу, в которой набор автономных ИИ агентов выполняет роли настоящей команды разработчиков — UX-дизайнера, бэкенд-инженера, full stack-разработчика и других специалистов. По данным компании, они обслуживают около 25 000 пользователей, включая бизнес-клиентов и непрофессиональных основателей, предоставляя возможность создавать приложения только по текстовому или голосовому описанию.

Например, ресторан может запросить «сделать систему бронирования столов с базой клиентов и управлением запасами». Altan агенты проектируют интерфейс, создают базу, пишут API, генерируют тесты и документацию — практически полностью без ручного вмешательства. При этом команда стартапа проверяет ключевые части кода и архитектуры, чтобы гарантировать надёжность и соответствие бизнес-логике.

2. Citigroup — корпоративная интеграция ИИ агентов

Citigroup запускает пилотную программу, где ИИ агенты выполняют сложные многоэтапные задачи с использованием моделей вроде Gemini (Google) и Claude (Anthropic). Агенты интегрированы через собственную платформу Citi и взаимодействуют с внутренними системами, что приближает использование к полной «агентной» парадигме.

3. Visma — европейская финтех группа

Visma использует Devin (Cognition AI) на платформе Azure для модернизации корпоративных приложений. По данным компании, применение ИИ агентов позволило увеличить продуктивность разработчиков в 2 раза и сократить затраты на 50%. Решение предоставляется как сервис партнёра, а не как собственный агент Visma.

4. Walmart — «супер агент» Wibey

Walmart разрабатывает и внедряет «супер агента» под названием Wibey, который управляет большим числом более простых ИИ агентов. Часть агентов автоматически выявляет баги (например, связанные с доступностью) и корректирует их, что повышает продуктивность инженеров. При этом Walmart — не AI стартап, а крупная технологическая корпорация, использующая агентов для внутренних процессов, а не для продажи как продукта.

5. Google — внутренние агенты разработки

В Google используется инструмент Jules — асинхронный ИИ агент, который способен исправлять баги, изменять файлы и готовить pull request'ы. По заявлению компании, их агенты справляются с многошаговыми задачами: генерация приложений, тестирование, обновление документации и деплой. Использование ИИ агентов повышает эффективность инженеров примерно на 10%.

Переход на AI Driven Development: текущая ситуация

Хотя концепция AI Driven Development вызывает огромный интерес, на практике большинство компаний пока только переходит на этот подход. Использование автономных ИИ агентов, способных самостоятельно писать код, тестировать и деплоить изменения, всё ещё находится на стадии пилотных проектов и экспериментов.

Крупные корпорации уже тестируют ИИ агентов в реальных рабочих процессах. Однако чаще всего агенты не работают полностью самостоятельно: решения ИИ проходят человеческое ревью, а задачи распределяются между человеком и агентом.

При этом многие малобюджетные стартапы используют ИИ агентов как способ компенсировать недостаток ресурсов:

- небольшие команды внедряют готовых ИИ агентов (например, GitHub Copilot Coding Agent или Devin AI) для автоматизации рутинных задач;
- агенты помогают писать код, генерировать тесты и управлять инфраструктурой, что позволяет стартапам ускорить разработку без найма большого числа разработчиков;
- для стартапов это особенно ценно, поскольку сокращает расходы и ускоряет выход продукта на рынок.

Основные причины осторожного внедрения у крупных компаний остаются:

- необходимость контроля качества и безопасности;
- ограниченная способность ИИ справляться с новыми, ранее не решёнными задачами;
- юридические и регуляторные вопросы;
- интеграция с существующими процессами разработки и корпоративной инфраструктурой.

Таким образом, на сегодняшний день большинство компаний используют гибридную модель: ИИ агенты берут на себя рутинные, повторяющиеся задачи, а человек остаётся стратегом и критически важным участником процесса. Перспектива полностью автономной разработки ещё пока реализуется лишь в отдельных пилотных проектах, но тенденция к расширению использования таких агентов в ближайшие годы очевидна.

Список литературы

1. [Хабр — «Agentic AI: мечта CEO или источник корпоративных уязвимостей?»](#)
2. [Хабр — «ИИ агенты: от теории к практике»](#)
3. [Forbes — «Agentic AI In Software Development: The New Standard For 2026»](#)
4. [Ведомости — «Агентский ИИ: новая парадигма искусственного интеллекта и место России в революции»](#)
5. [World Journal of Advanced Engineering Technology & Sciences — «Agentic AI in the Age of Hyper Automation»](#)
6. [The Wall Street Journal — «AI Agents Arrive at Citigroup»](#)
7. [Forbes — «The Rise of the Agentic SDLC: How AI Agents Are Redefining Software Development»](#)
8. [Business Insider — «Altan: Startup Using Agentic AI to Build Apps Autonomously»](#)
9. [arXiv — «AgentMesh: Multi-Agent Framework for Autonomous Software Development»](#)