

## ЭВОЛЮЦИЯ ИИ-АГЕНТОВ В ПРОГРАММИРОВАНИИ Evolution of AI agents for programming

**Е. Ю. Бубнова**, студент

**К. Н. Иванов**, студент

**С. В. Кушуков**, студент

Поволжский государственный университет телекоммуникаций и информатики  
(Самара, Московское шоссе, 77)

### Аннотация

Статья посвящена анализу трансформации роли искусственного интеллекта в индустрии программного обеспечения, от вспомогательных инструментов до автономных ИИ-агентов. Рассматриваются ключевые тенденции последних двух десятилетий, включая развитие крупных языковых моделей (LLM) и их влияние на автоматизацию задач разработки. Особое внимание уделяется примерам практической реализации автономных агентов, таких как GitHub Copilot и SWE-agent, а также вопросам безопасности, этики и правового регулирования, возникающим в связи с их внедрением.

**Ключевые слова:** ИИ-агенты, разработка ПО, автоматизация, нейросети, коллаборация, авторское право.

### Summary

The article is devoted to the analysis of the transformation of the role of artificial intelligence in the software industry, from auxiliary tools to autonomous AI agents. The key trends of the last two decades are considered, including the development of large-scale language models (LLM) and their impact on the automation of development tasks. Special attention is paid to examples of practical implementation of autonomous agents such as GitHub Copilot and SWE-agent, as well as security, ethics and legal regulation issues arising from their implementation.

**Keywords:** AI agents, software development, automation, neural networks, collaboration, copyright.

*Работа выполнена при финансовой поддержке РФФИ (грант № 23–61–10032)*

Современная индустрия программного обеспечения переживает качественный сдвиг: если ранее искусственный интеллект применялся преимущественно в качестве вспомогательного инструмента, ускоряющего отдельные этапы разработки, то в настоящее время наблюдается активное внедрение автономных ИИ-агентов, способных самостоятельно создавать, тестировать и оптимизировать программный код. Этот переход свидетельствует не только о технологической эволюции, но и о фундаментальном переосмыслении роли человека в процессе разработки программных решений. На фоне роста сложности цифровых систем и необходимости ускоренного вывода продуктов на рынок автономные ИИ-агенты становятся ключевыми участниками инженерных процессов.

В течение последних двух десятилетий наблюдается устойчивый рост применения технологий искусственного интеллекта в области программной инженерии. На начальном этапе их функциональность ограничивалась автоматизацией рутинных операций: автодополнением кода, статическим анализом, генерацией документации. Однако с появлением архитектур глубокого обучения и развитием крупных языковых моделей (LLM), таких как GPT, Codex, CodeGen, функциональные возможности ИИ существенно расширились. Эти системы получили

способность не просто поддерживать, но и частично заменять разработчиков в выполнении критически важных задач программирования. В связи с этим особую значимость приобретает анализ перехода от вспомогательных инструментов, встроенных в среды разработки, к автономным агентам, выполняющим функции, характерные для архитекторов, программистов, тестировщиков и даже менеджеров цифровых продуктов.

Целью настоящей статьи является анализ трансформации ИИ-агентов в контексте разработки программного обеспечения, выявление их влияния на производственные процессы и систематизация ключевых вызовов, сопровождающих внедрение автономных систем. В рамках исследования ставятся следующие задачи: классификация типов ИИ-агентов и их функциональных особенностей; оценка влияния агентов на производительность команд и бизнес-процессы; анализ технических, юридических и этических рисков, связанных с автономией систем.

ИИ-агенты представляют собой программные системы, обладающие способностью к автономному выполнению задач, включая взаимодействие с программными интерфейсами (API), средствами разработки (IDE), управление контекстом, планирование и обучение на основе опыта [1]. В отличие от традиционных LLM, агенты включают механизмы долговременной памяти, управления состоянием, а также возможности для выполнения многозадачной деятельности в рамках сложной цифровой среды. Согласно классификации, предложенной IBM, выделяют пять типов ИИ-агентов:

1. Рефлексные агенты, реагирующие на входные данные без использования модели окружающей среды.
2. Модель-ориентированные агенты, формирующие представление о мире и учитывающие предыдущее состояние при принятии решений.
3. Целевые агенты, действующие на основе заранее заданных целей и механизмов планирования.
4. Утилитарные агенты, максимизирующие функцию полезности с учетом условий неопределенности; обучающиеся агенты, использующие машинное обучение для адаптации поведения и оптимизации деятельности [2].

Примером реализации целевого агента является GitHub Copilot, предлагающий контекстуально релевантные фрагменты кода, автоматическое дополнение и рефакторинг. С введением режима Agent Mode в 2025 году, функциональность Copilot была расширена: агент может выполнять многошаговые задачи — анализ ошибок, генерация исправлений, выполнение тестов и формулирование терминальных команд [3]. Другой пример — обучающийся агент SWE-agent, разработанный исследовательской группой Принстонского университета. Он способен выполнять сложные задачи: отладку, исправление и анализ кода в автономном режиме, взаимодействуя с внешними инструментами через специализированный API-интерфейс [4].

Автоматизация рутинных операций является одной из ключевых сфер применения ИИ-агентов. По мере развития моделей, таких как Copilot Edits, появилась возможность многофайлового редактирования исходного кода по текстовому описанию задачи. Это реализуется посредством семантического анализа структуры проекта и логического сопоставления контекста. В режиме Agent Mode Copilot способен выполнять функции, выходящие за рамки традиционного помощника: запуск тестов, исправление ошибок, управление зависимостями и конфигурацией среды. Подобные функции ранее выполнялись группой специалистов — разработчиком, DevOps-инженером, тестировщиком. ИИ-агенты также способствуют ускорению обучения новых специалистов. Встраиваясь в IDE, они предоставляют интерактивные подсказки, адаптированные под текущую задачу. Согласно данным Microsoft Research (2023), внедрение

подобных инструментов позволяет сократить среднее время адаптации разработчика к новым технологиям на 30–40% [5].

Коллаборативные возможности агентов особенно выражены в режимах «парного программирования», когда агент участвует в анализе архитектуры проекта, выявлении узких мест и предложении архитектурных решений. В ряде случаев, как показали эксперименты с использованием SWE-agent, агент может инициировать разбиение задачи на подзадачи, передавая часть исполнения другим программным компонентам.

Практические данные подтверждают эффективность ИИ-инструментов: по отчёту 2023 года McKinsey & Company, компании, внедрившие ИИ-агентов, фиксируют рост скорости вывода новых функций на рынок до 50% и снижение частоты ошибок на продакшн-среде до 25% [6].

С увеличением степени автономии ИИ-агентов особое значение приобретают механизмы контроля, обеспечивающие безопасность и управляемость. Практика «человека в цикле» (human-in-the-loop) становится необходимостью, особенно при выполнении операций с доступом к конфиденциальной информации. Так, GitHub внедрил систему Copilot Control System, позволяющую пользователям управлять уровнем доступа к данным и ограничивать передачу контекста третьим сторонам. На международном уровне обсуждаются юридические аспекты авторства кода, сгенерированного ИИ. В США подан коллективный иск против Microsoft, GitHub и OpenAI, связанный с использованием защищенного авторским правом кода для обучения Copilot. Согласно разъяснениям Бюро авторского права США, объекты, созданные исключительно ИИ, не подлежат авторской защите, если в их создании отсутствовал творческий вклад человека [7].

Кроме юридических, встают и этические вопросы. Расширение функций агентов может привести к смещению акцентов в деятельности разработчиков: от творческого участия — к контролю и валидации работы ИИ. Это, в свою очередь, ставит под вопрос долгосрочные перспективы человеческого вклада в инженерный процесс и требует выработки новых профессиональных стандартов.

Перспективы развития ИИ-агентов связаны с дальнейшим увеличением уровня автономности. Разрабатываются системы, способные генерировать не отдельные функции, а полноценные программные модули с архитектурной структурой, документацией и тестами (например, GitHub Project Padawan). Ожидается глубокая интеграция ИИ-агентов с мультимодальными языковыми моделями нового поколения (Gemini, GPT-5), что позволит им обрабатывать текстовые, визуальные и аудиоданные в едином контексте. Это откроет новые возможности в создании комплексных интерфейсов, взаимодействующих с пользователем на разных уровнях. Кроме того, формируется экосистема кастомизируемых агентов. Платформы, подобные GPT Store, уже предлагают маркетплейсы, где пользователи могут создавать, распространять и применять специализированных агентов под конкретные задачи разработки.

ИИ-агенты становятся неотъемлемой частью современного жизненного цикла разработки программного обеспечения. Их внедрение способствует повышению производительности, ускорению адаптации специалистов и улучшению качества инженерных решений. Однако рост автономности несет с собой риски — технические, юридические и этические, которые требуют системного подхода. Ключевыми направлениями научных исследований в данной области становятся: разработка механизмов прозрачности принятия решений ИИ, обеспечение юридической ответственности за действия агентов и формирование международных этических норм и стандартов. Только в условиях контролируемой и ответственной интеграции ИИ-агенты смогут стать устойчивым элементом цифровой инженерии будущего.

## Библиографический список

1. AI-агенты для автоматизации задач: Как оператор OpenAI меняет подходы к кодированию и бронированию? // External Software. URL: <https://external.software/archives/17393>.
2. Агентный подход к ИИ // AI Lectures. URL: <https://voronina-ai.github.io/docs/I-15/>.
3. Agent mode: available to all users and supports MCP // Visual Studio Code. URL: <https://code.visualstudio.com/blogs/2025/04/07/agentMode>.
4. SWE-Agent: Инновационный пайплайн для -4 по исправлению багов на GitHub // Analytic Vidhya. URL: <https://www.analyticsvidhya.com/blog/2024/04/swe-agent-the-latest-open-source-ai-powered-software-developer/>.
5. Отчет Microsoft о новом будущем работы за 2023 год // Microsoft – Национальный портал в сфере искусственного интеллекта. URL: [https://ai.gov.ru/knowledgebase/obrazovanie-i-kadryii/2024\\_otchet\\_microsoft\\_o\\_novom\\_buduschem\\_raboty\\_za\\_2023\\_god\\_microsoft\\_new\\_future\\_of\\_work\\_report\\_2023\\_microsoft/](https://ai.gov.ru/knowledgebase/obrazovanie-i-kadryii/2024_otchet_microsoft_o_novom_buduschem_raboty_za_2023_god_microsoft_new_future_of_work_report_2023_microsoft/).
6. Корпоративная стратегия и искусственный интеллект // McKinsey – Национальный портал в сфере искусственного интеллекта. URL: [https://ai.gov.ru/knowledgebase/votraslyakh/2023\\_iskusstvennyu\\_intellekt\\_v\\_strategii\\_artificial\\_intelligence\\_in\\_strategy\\_mckinsey/](https://ai.gov.ru/knowledgebase/votraslyakh/2023_iskusstvennyu_intellekt_v_strategii_artificial_intelligence_in_strategy_mckinsey/).
7. В суде США разваливается коллективный иск «разработчики против Microsoft, GitHub и OpenAI из-за работы GitHub Copilot» // Хабр. URL: <https://habr.com/ru/news/827836/>.