

УДК 004.8

***ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ИНСТРУМЕНТАХ РАЗРАБОТЧИКА:
GITHUB COPILOT И ТРАНСФОРМАЦИЯ ПРОЦЕССА
ПРОГРАММИРОВАНИЯ***

Кобец А.Р.

магистрант,

Сочинский государственный университет,

Сочи, Россия

Салова Т.Л.

кандидат технических наук, доцент,

кафедра Информационных технологий и математики,

Сочинский государственный университет,

Сочи, Россия

Аннотация

В статье рассматривается влияние искусственного интеллекта на процесс разработки программного обеспечения на примере GitHub Copilot. Искусственный интеллект становится неотъемлемой частью инструментария современного разработчика, кардинально меняя подходы к программированию и автоматизируя рутинные задачи. В работе анализируются технологические основы GitHub Copilot, базирующегося на модели OpenAI Codex, и его влияние на производительность разработчиков. Представлены результаты исследований, демонстрирующих увеличение скорости разработки до 55% при использовании ИИ-ассистентов. Рассматриваются ключевые аспекты трансформации процесса программирования: изменение роли разработчика от написания кода к управлению его генерацией и

верификацией, автоматизация рутинных операций и появление новых подходов к обучению. Также обсуждаются вызовы и перспективы развития технологии, включая вопросы качества генерируемого кода, безопасности и этические аспекты применения ИИ в разработке программного обеспечения.

Ключевые слова: искусственный интеллект, GitHub Copilot, разработка программного обеспечения, машинное обучение, автоматизация кода, производительность разработчиков, трансформация процессов.

ARTIFICIAL INTELLIGENCE IN DEVELOPER TOOLS: GITHUB COPILOT AND TRANSFORMATION OF THE PROGRAMMING PROCESS

Kobec A.R

undergraduate,

Sochi State University,

Sochi, Russia

Salova T.L.

candidate of Technical Sciences, an associate professor,

Department of Information Technologies and Mathematics,

Sochi State University,

Sochi, Russia

Abstract

The article examines the impact of artificial intelligence on the software development process using GitHub Copilot as an example. Artificial intelligence is becoming an integral part of the modern developer's toolkit, radically changing approaches to programming and automating routine tasks. The paper analyzes the technological

foundations of GitHub Copilot, based on the OpenAI Codex model, and its impact on developer productivity. The results of studies demonstrating an increase in development speed of up to 55% when using AI assistants are presented. The key aspects of the transformation of the programming process are considered: changing the role of the developer from writing code to managing its generation and verification, automating routine operations and the emergence of new approaches to learning. The challenges and prospects for the development of technology are also discussed, including issues of the quality of generated code, security and ethical aspects of using AI in software development.

Key words: artificial intelligence, GitHub Copilot, software development, machine learning, code automation, developer productivity, process transformation.

Современная разработка программного обеспечения переживает период фундаментальной трансформации, обусловленной активным внедрением технологий искусственного интеллекта. Если раньше программирование оставалось исключительно человеческой деятельностью, требующей глубоких знаний алгоритмов и структур данных, то сегодня интеллектуальные системы способны не только помогать разработчикам, но и самостоятельно писать, оптимизировать и исправлять программный код [1]. Развитие технологий машинного обучения и глубоких нейронных сетей позволило создать мощные инструменты, которые значительно упрощают и ускоряют процесс разработки [2].

Одним из наиболее заметных примеров применения искусственного интеллекта в программировании стал GitHub Copilot - ИИ-ассистент для разработчиков, представленный компанией GitHub в 2021 году [3]. Этот инструмент базируется на технологии OpenAI Codex и использует возможности больших языковых моделей для генерации программного кода на основе

комментариев на естественном языке или контекста уже написанного кода [4]. По данным исследований, около 30% кода на платформе GitHub уже создается при помощи Copilot, что свидетельствует о масштабном проникновении ИИ-технологий в процесс разработки программного обеспечения [5].

Целью настоящего исследования является анализ влияния GitHub Copilot на трансформацию процесса программирования и оценка эффективности применения искусственного интеллекта в разработке программного обеспечения.

GitHub Copilot как инструмент разработки

GitHub Copilot представляет собой ИИ-ассистент, который интегрируется в популярные среды разработки и предоставляет разработчикам интеллектуальные подсказки и автодополнение кода в режиме реального времени [3]. В основе работы инструмента лежит модель OpenAI Codex - усовершенствованная версия GPT-3, обученная на миллиардах строк открытого исходного кода из репозитория GitHub [4]. Технология использует методы глубокого обучения для понимания контекста задачи и генерации релевантных фрагментов кода.

Принцип работы GitHub Copilot основан на анализе контекста: система изучает уже написанный код, комментарии программиста и структуру проекта, после чего предлагает варианты продолжения [6]. Инструмент поддерживает множество языков программирования, включая Python, JavaScript, TypeScript, Ruby, Go и другие, что делает его универсальным решением для различных типов проектов. Интеграция с популярными IDE (Visual Studio Code, JetBrains IDEs, Neovim) обеспечивает бесшовный рабочий процесс и позволяет разработчикам использовать ИИ-помощника без изменения привычных рабочих процессов [7].

Важной особенностью GitHub Copilot является способность генерировать не только отдельные строки кода, но и целые функции, классы и даже модули на основе текстового описания задачи [4]. Система также может предлагать

альтернативные варианты реализации, что способствует улучшению качества кода и расширению кругозора разработчиков. При этом генерируемый код сопровождается комментариями, что улучшает его читаемость и поддерживаемость.

Влияние на производительность разработчиков

Многочисленные исследования демонстрируют значительное влияние GitHub Copilot на производительность разработчиков. Согласно исследованию, проведенному командой GitHub, разработчики, использующие Copilot, выполняют задачи на 55% быстрее по сравнению с контрольной группой, работающей без ИИ-ассистента [8]. Анализ выборки из более чем 930 тысяч пользователей показал, что в среднем разработчики принимают около 30% предложений кода от ИИ-помощника, причем этот показатель растет по мере знакомства с инструментом.

Исследования также выявили, что наибольшую пользу от использования GitHub Copilot получают начинающие разработчики и специалисты, работающие с незнакомыми технологиями [8]. ИИ-ассистент помогает им быстрее осваивать новые языки программирования и фреймворки, предоставляя примеры корректного использования синтаксиса и паттернов проектирования. Для опытных разработчиков основное преимущество заключается в автоматизации рутинных задач и ускорении написания шаблонного кода.

Вместе с тем, исследования фиксируют и определенные вызовы. Некоторые работы указывают на увеличение количества ошибок в коде при использовании ИИ-ассистентов, что требует более тщательной проверки и тестирования сгенерированного кода. Это подчеркивает важность критического подхода к использованию инструментов на базе искусственного интеллекта и необходимость сохранения человеческого контроля над процессом разработки.

Исследование GitHub показало, что от 60 до 75% разработчиков сообщают о повышении удовлетворенности своей работой при использовании Copilot, меньшем разочаровании при написании кода и возможности сосредоточиться на более творческих аспектах разработки [8]. Разработчики отмечают, что ИИ-ассистент помогает им оставаться в состоянии потока (73%) и экономить ментальную энергию при выполнении повторяющихся задач (87%).

Трансформация процесса программирования

Внедрение инструментов на базе искусственного интеллекта приводит к фундаментальной трансформации процесса программирования [1]. Роль разработчика постепенно смещается от непосредственного написания кода к управлению его генерацией, верификации и архитектурному проектированию. ИИ-ассистенты берут на себя рутинные операции: написание шаблонного кода, создание тестов, генерацию документации, что позволяет разработчикам фокусироваться на решении более сложных задач и стратегических аспектах проектов [2].

Автоматизация рутинных задач с помощью ИИ приводит к ускорению разработки программного обеспечения в среднем на 50%, что особенно заметно при выполнении таких операций, как написание простых тестов или рефакторинг кода [1]. Это позволяет компаниям сокращать время вывода продуктов на рынок и более эффективно реагировать на изменяющиеся требования бизнеса.

Появление ИИ-ассистентов также меняет подходы к обучению программированию [2]. Новички получают возможность учиться на примерах кода, генерируемого искусственным интеллектом, что помогает быстрее осваивать лучшие практики и паттерны программирования. Однако это создает риск поверхностного понимания базовых концепций, если обучающиеся полагаются исключительно на ИИ без глубокого изучения фундаментальных принципов.

Трансформация затрагивает и командную работу: ИИ-инструменты способствуют стандартизации кода, улучшению его читаемости и снижению технического долга [3]. Системы на базе искусственного интеллекта могут анализировать исторические данные проектов, выявлять уязвимости и предлагать оптимизации, что повышает общее качество разрабатываемого программного обеспечения.

Перспективы и вызовы

Развитие искусственного интеллекта в сфере разработки программного обеспечения открывает широкие перспективы для дальнейших инноваций. Ожидается, что к 2030 году инструменты на базе генеративного ИИ смогут увеличить производительность эквивалента 15 миллионов разработчиков, что приведет к росту мирового ВВП более чем на 1,5 триллиона долларов. Российские компании также активно внедряют технологии искусственного интеллекта: по данным исследований, около 20% отечественных ИТ-компаний уже используют генеративный ИИ при разработке программного обеспечения [1].

Перспективными направлениями развития являются создание специализированных ИИ-ассистентов для конкретных доменов, улучшение понимания контекста и бизнес-логики приложений, а также интеграция с системами автоматизированного тестирования и непрерывной интеграции [2]. Развитие российских альтернатив зарубежным ИИ-ассистентам становится особенно актуальным в условиях технологической независимости.

Вместе с тем, существуют серьезные вызовы, требующие внимания. Остаются открытыми вопросы безопасности генерируемого кода, защиты интеллектуальной собственности и этического использования ИИ. Зависимость от зарубежных платформ создает риски для критически важных проектов, что подчеркивает необходимость развития отечественных решений.

Следует подчеркнуть, что ИИ-ассистенты требуют от разработчиков новых компетенций: умения эффективно формулировать запросы, критически оценивать генерируемый код и интегрировать автоматизированные решения в рабочий процесс.

Заключение

Искусственный интеллект в инструментах разработчика, таких как GitHub Copilot, знаменует собой начало новой эры в программировании, где автоматизация рутинных задач и интеллектуальная поддержка становятся неотъемлемой частью рабочего процесса. Исследования демонстрируют значительное повышение производительности разработчиков и улучшение качества кода при правильном использовании ИИ-ассистентов. Трансформация процесса программирования проявляется в изменении роли разработчика, новых подходах к обучению и оптимизации рабочих процессов.

Практическая значимость применения искусственного интеллекта в разработке программного обеспечения заключается в ускорении создания продуктов, снижении затрат и повышении доступности программирования для широкого круга специалистов.

Для успешного внедрения этих технологий необходимо развитие соответствующих компетенций у разработчиков, решение вопросов безопасности и создание отечественных ИИ-решений для обеспечения технологической независимости. Дальнейшие исследования должны быть направлены на изучение долгосрочного влияния ИИ-инструментов на качество программного обеспечения и профессиональное развитие разработчиков.

Библиографический список

1. Польский Г.А. Тенденции в разработке программного обеспечения с использованием искусственного интеллекта // Инновации. Наука. Образование. 2022. № 53. С. 1486-1491.
2. Как искусственный интеллект помогает разрабатывать программное обеспечение. - URL: <https://www.tadviser.ru/index.php> (дата обращения: 29.10.2025).
3. Проблема искусственно сгенерированных исходных кодов программного обеспечения // Научные исследования. 2022. № 1. С. 30–39. – URL: <https://elibrary.ru/item.asp?id=49525450> (дата обращения: 14.12.2025).
4. Шабунов А.В. Использование инструментов машинного обучения в задаче прогнозирования // Прикладная информатика. 2023. № 18. С. 20–45. – URL: <https://elibrary.ru/item.asp?id=53843946> (дата обращения: 14.12.2025).
5. Настройка GitHub Copilot в вашей среде разработки. - URL: <https://docs.github.com/ru/copilot/how-tos/configure-personal-settings/configure-in-ide> (дата обращения: 29.10.2025).
6. Алексеев Д.В., Суздальцев А.И. Использование генеративного ИИ при проектировании цифровых решений для отраслей промышленности // Экономика промышленности. 2024. Т. 17, № 4. С. 407–426. – URL: <https://elibrary.ru/item.asp?id=62673177> (дата обращения: 19.11.2025).
7. Кеда Н.А., Малиновская О.В. Применение методов искусственного интеллекта в цифровой трансформации бизнес-процессов // Вестник Алтайской академии экономики и права. 2024. № 2-1. С. 59–65. – URL: <https://elibrary.ru/item.asp?id=75990098> (дата обращения: 14.12.2025).
8. Алферов В.Н., Лысенко В.С. Влияние искусственного интеллекта на управление организацией // Инновации. Наука. Образование. 2025. № 80. С. 244–250. – URL: <https://elibrary.ru/item.asp?id=80458554> (дата обращения: 19.11.2025).