

Панюкова Екатерина Владимировна
Самарский государственный технический университет

Интеграция технологий искусственного интеллекта в процесс подготовки бакалавров программной инженерии

Аннотация. В статье рассматривается проблема интеграции технологий генеративного искусственного интеллекта (на примере больших языковых моделей) в процесс подготовки бакалавров по направлению «Программная инженерия». Актуальность обусловлена стремительным внедрением ИИ-инструментов (ChatGPT, GitHub Copilot) в индустрию разработки ПО, что создаёт противоречие между требованиями работодателей и рисками деградации фундаментальных компетенций студентов. Автором выявлены основные риски: когнитивная лень, снижение алгоритмического мышления, академическое мошенничество. Предложена дифференцированная модель интеграции ИИ в учебный процесс в зависимости от дисциплины: полное исключение на младших курсах (алгоритмизация), контролируемое использование на старших (проектирование архитектуры, тестирование). Сделан вывод о необходимости формирования у студентов метакогнитивных навыков и ответственного использования ИИ как инструмента, расширяющего, но не замещающего инженерное мышление.

Ключевые слова: искусственный интеллект, генеративные нейросети, программная инженерия, бакалавриат, когнитивная лень, методическая модель, ChatGPT, GitHub Copilot, высшее образование, цифровая трансформация.

Panyukova Ekaterina Vladimirovna
Samara State Technical University

Integration of artificial intelligence technologies into the training process of bachelor's degree students in software engineering

Abstract. The article examines the problem of integrating generative artificial intelligence technologies (using large language models as an example) into the training process of bachelor's students in the field of "Software Engineering." The relevance is driven by the rapid adoption of AI tools (ChatGPT, GitHub Copilot) in the software development industry, which creates a contradiction between employer demands and the risks of degrading students' fundamental competencies. The author identifies the main risks: cognitive laziness, a decline in algorithmic thinking, and academic dishonesty. A differentiated model for AI integration depending on the discipline is proposed: complete exclusion in the early years (algorithmization) and controlled use in the later years (architecture design, testing). The conclusion is drawn that it is necessary to develop students' metacognitive skills and responsible use of AI as a tool that expands, but does not replace, engineering thinking.

Keywords: artificial intelligence, generative neural networks, software engineering, bachelor's degree program / undergraduate studies, cognitive laziness, methodological model, ChatGPT, GitHub Copilot, higher education, digital transformation.

Введение

Актуальность исследования. Стремительное развитие генеративных нейросетей, таких как ChatGPT, GitHub Copilot, и других, в период 2022–2026 годов привело к фундаментальным изменениям в индустрии разработки программного обеспечения. Если ранее искусственный интеллект (ИИ) использовался преимущественно для решения

узкоспециализированных задач (распознавание образов, анализ данных), то современные большие языковые модели (LLM) способны генерировать программный код, писать тесты, оптимизировать код, документировать, а также проектировать архитектуру приложений по текстовому описанию. Согласно исследованиям GitHub (2024), до 46% кода в проектах, использующих Copilot, пишется с помощью ИИ, а время выполнения типовых задач сокращается на 55%. В этих условиях работодатели ожидают от выпускников программной инженерии не только классических знаний, но и умения эффективно взаимодействовать с ИИ-ассистентами, критически оценивать их результаты и интегрировать их в производственный процесс.

Проблема исследования. Использование инструментов искусственного интеллекта при разработке программных артефактов создает проблемы в системе подготовки студентов. Активное использование генеративных моделей ИИ в учебном процессе сказывается на формировании базовых инженерных компетенций. К основным проблемам можно отнести снижение уровня алгоритмического мышления из-за склонности делегировать вычислительные задачи ИИ, невозможность самостоятельной отладки кода, что влечет рост числа ошибок в машинном коде. В таких условиях высшая школа сталкивается с важной педагогической задачей по нахождению оптимального баланса между использованием инструментов ИИ в образовательном процессе и формированием у студентов профессиональных навыков в области разработки программного обеспечения.

Цель статьи - обосновать интеграцию технологий ИИ (на примере генеративных нейросетей) в процесс подготовки бакалавров по направлению «Программная инженерия», позволяющую снять выявленное противоречие и повысить качество формирования профессиональных компетенций.

Объект исследования - процесс профессиональной подготовки бакалавров программной инженерии в условиях цифровой трансформации высшего образования.

Предмет исследования — методы, формы и средства интеграции технологий ИИ (генеративных нейросетей для программирования) в содержание, организационные формы и оценочные средства дисциплин профессионального цикла.

Характеристика исследований в области ИИ и программной инженерии

В последние годы наблюдается значительный рост интереса к интеграции технологий ИИ в процесс подготовки IT-специалистов. Согласно обзору [1], проведенному на основе анализа 40 рецензируемых исследований в области программной инженерии, отмечается большое количество публикаций по теме использования ИИ в обучении программированию. Авторы обзора выделяют шесть ключевых направлений в использовании ИИ: анализ и прогнозирование успеваемости обучающихся, выявление «групп риска», разработка контента, генерация подсказок и рекомендаций, а также автоматическое извлечение учебного контента.

Другой обзор, охвативший 58 рецензируемых исследований за период 2022–2025 годов, фокусируется непосредственно на применении ИИ-агентов в обучении программированию [2]. Исследователи выделили три основные категории ИИ-инструментов: чат-боты, генеративные нейросети и интеллектуальные обучающие системы, причем генеративные модели оказались наиболее часто изучаемыми инструментами ИИ.

Среди ключевых преимуществ использования ИИ отмечаются персонализированная обратная связь, улучшение учебных результатов и экономия времени преподавателей.

Однако авторы фиксируют и существенные вызовы: сложности внедрения (отмечены в 93,1% исследований), риск чрезмерной зависимости студентов от ИИ, приводящий к поверхностному обучению (65,52% исследований), а также проблемы с ошибками ИИ и интеллектуальной честностью.

Риски интеграции генеративного ИИ в подготовку бакалавров программной инженерии

Традиционное понимание плагиата как заимствования текста или кода без указания авторства сталкивается с новыми вызовами при использовании генеративных нейросетей при обработке контекста учебных и научных источников.

Программный код, сгенерированный генеративной нейросетью (например, ChatGPT) в отличие от копирования готового решения из открытого источника (например, GitHub) часто является уникальной комбинацией фрагментов из тысяч обучающих примеров. Это создает «серую зону», т.е. такой код формально не является прямой копией, но и не может считаться полностью самостоятельной работой студента.

Согласно исследованию [3] - до 50% кода, сгенерированного ИИ для учебных задач, требовало коррекции с участием человека. Однако в условиях отсутствия контроля студенты могут сдавать сгенерированный код без какой-либо модификации, выдавая его за свою разработку. При этом существующие детекторы ИИ демонстрируют низкую точность на программном коде, особенно после переработки.

Традиционные формы контроля (проверка домашних заданий, автоматическое тестирование кода) становятся недостаточными. Как указывают авторы работы [4], студенты очень часто используют инструменты ИИ при выполнении учебных заданий, что негативно сказывается на качестве и результате их обучения.

К числу рисков использования ИИ относят феномен «когнитивная лень» или фигурирующий в литературе под названием «метакогнитивная лень». Данное понятие отражает склонность обучающихся безоговорочно опираться на готовые алгоритмические решения, что закономерно снижает критичность восприятия и глубину анализа данных. В статье «Beware of Metacognitive Laziness: Effects of Generative Artificial Intelligence on Learning Motivation, Processes, and Performance» авторы дополнительно подчёркивают, что «генеративные модели (в частности, ChatGPT) способны формировать технологическую зависимость и выступать катализатором указанной «метакогнитивной лени» [6].

Можно сделать вывод, что термин «метакогнитивная лень» обозначает систематическую практику делегирования нейросетям мыслительные функции, которые традиционно служили основой для становления профессионального мышления инженера-программиста.

Можно выделить основные проявления «когнитивной лени» в учебном процессе при изучении дисциплин направлений программной инженерии:

- а) В области алгоритмизации.
 - Студент перестает самостоятельно проектировать алгоритмы даже средней сложности, сразу обращаясь к ИИ.
 - Снижается способность оценивать временную и пространственную сложность алгоритмов, так как ИИ генерирует «готовое» решение.
 - Утрачивается навык декомпозиции задачи на подзадачи - ИИ делает это автоматически.
- б) В области отладки программ.
 - Студент не развивает навык чтения и понимания «чужого» (например, сгенерированного) кода.
 - Вместо анализа стек-трейса и логических ошибок студент копирует ошибку в чат-бот и получает рекомендации по исправлению ошибок.
 - Формируется «слепая зависимость» - без ИИ студент не может найти и исправить даже простую синтаксическую ошибку.
- в) В области проектирования архитектуры ПО.
 - ИИ предлагает архитектурные решения (паттерны, структуру классов, зоны ответственности компонентов ПО), но студент не понимает выбор таких решений.
 - Студент не может обосновать архитектурный выбор элементов ПО и произвести сравнение альтернативных решений.

Авторы [2] отмечают, что поверхностное обучение (shallow learning) становится прямым следствием некритического использования генеративных нейросетей. Студенты

получают «иллюзию компетентности»: они могут объяснить, что делает код (поскольку ИИ предоставил комментарии или документацию), но не могут написать аналогичный код с нуля или модифицировать его для решения измененной задачи.

Исследование [3] подтверждает данный вывод - около 50% сгенерированного ИИ кода требовало человеческой коррекции, однако студенты, не обладающие достаточными базовыми навыками, не способны идентифицировать и исправить данные ошибки. В результате они сдают работы, содержащие логические, стилистические и даже синтаксические ошибки, которые ИИ «замаскировал» под корректное решение.

Риски плагиата и когнитивной лени тесно взаимосвязаны. У студента, привыкшего полагаться на ИИ для генерации решений, не сформирована внутренняя мотивация к самостоятельному выполнению заданий. И если студент сознательно использует ИИ для плагиата, то со временем утрачивает навыки самостоятельного решения задач.

Интеграция ИИ в образовательный процесс подготовки специалистов по программной инженерии

ИИ в настоящее время способен органично встраиваться во все ключевые этапы жизненного цикла разработки программного обеспечения (SDLC), кардинально меняя подходы к выполнению рутинных и творческих задач. Этот процесс начинается уже с фазы определения требований - ИИ может помогать анализировать запросы заказчиков, выявлять противоречия и неполноту спецификаций требований.

На этапе проектирования ИИ-инструменты способны генерировать варианты архитектурных решений, предлагать оптимальные паттерны проектирования и автоматически создавать диаграммы классов, последовательностей или развертывания.

При разработке ПО такие задачи как автодополнение или рефакторинг кода, обнаружение потенциальных уязвимостей становятся стандартной функциональностью ИИ.

На этапе документирования ИИ помогает генерировать описания API, инструкции по развертыванию и комментарии к коду, анализировать корректность формулировки требований к ПО. Наконец, в тестировании ИИ применяется для автоматического создания тестовых сценариев, генерации тестовых данных, предсказания зон риска и даже самостоятельного прогона регрессионных тестов.

Тем не менее, при подготовке бакалавров по направлению программной инженерии необходимо установить четкие правила использования инструментов ИИ в образовательном процессе. Искусственному интеллекту можно делегировать только те задачи, которые не оказывают влияние на формирование базовых профессиональных компетенций, а также позволяют оптимизировать рутинные операции.

Ключевая цель образования в области подготовки программной инженерии - сформировать у студентов фундаментальное понимание принципов разработки ПО, алгоритмическое мышление, навыки критического анализа и проектирования архитектурных решений. Делегирование задач ИИ на этапе обучения ведет к риску формирования поверхностных компетенций, когда студент умеет только формулировать запросы к модели, но не понимает основных принципов и методов проектирования и разработки ПО. В рамках обоснованных сценариев применения ИИ можно отнести решение конкретных прикладных задач, которые не замещают, а дополняют традиционные методы обучения.

Рассмотрим их на примере некоторых дисциплин по направлению «Программная инженерия».

Дисциплины «Алгоритмизация и программирование», «Алгоритмы и структуры данных» являются фундаментом подготовки. Именно здесь формируются базовые навыки алгоритмического мышления, которые в наибольшей степени подвержены риску снижения при неконтролируемом использовании ИИ. Поэтому использование инструментов ИИ в данных дисциплинах не должно быть встроено в учебный процесс.

В дисциплинах «Проектирование архитектуры ПО», «Объектно-ориентированный анализ и проектирование ПО» ИИ может выступать как инструмент генерации визуальных моделей процессов (нотации BPMN или IDEF0) и архитектурных решений на основе требований к ПО или по текстовому описанию (нотация С4). Использование ИИ возможно только в том случае, когда студент владеет базовыми навыками использования инструментов анализа и проектирования ПО.

В дисциплинах «Верификация и валидация ПО», «Качество программного обеспечения» искусственный интеллект может использоваться для генерации шаблонов тест-кейсов (например, для модульного или интеграционного тестирования), тестовых данных. Сгенерированные тестовые артефакты могут не соответствовать основным свойствам хорошей тестовой документации. Поэтому финальная валидация тестов, понимание граничных условий и умение модифицировать сгенерированный код под конкретный контекст остаются обязательными компетенциями, формируемыми без искусственного интеллекта.

Таким образом, интеграция инструментов искусственного интеллекта в образовательный процесс бакалавров программной инженерии должна строиться на принципе «расширения возможностей, а не замещения мышления».

Искусственный интеллект выступает в роли мощного вспомогательного инструмента для ускорения рутинных операций (генерация диаграмм, черновиков тестов, типовых фрагментов документации), но все этапы анализа, проектирования и принятия критических решений остаются исключительно за человеком, проходящим полноценную инженерную подготовку.

Заключение

Исследование позволило выявить противоречие в современной подготовке бакалавров программной инженерии - между объективной потребностью рынка в выпускниках, владеющих инструментами генеративного ИИ, и высокими рисками снижения у студентов фундаментальных профессиональных компетенций (алгоритмическое мышление, отладка, архитектурное проектирование и т.д.) при бесконтрольном использовании этих технологий.

Проанализированы основные риски интеграции ИИ - «когнитивная лень» (метакогнитивная лень), ведущая к поверхностному обучению, снижение критичности мышления. Обоснован дифференцированный подход к внедрению ИИ в зависимости от дисциплин подготовки - полное исключение ИИ или строгое ограничение на младших курсах (фундаментальные алгоритмические дисциплины), контролируемое использование ИИ в качестве вспомогательного инструмента на старших курсах (проектирование архитектуры, тестирование, документирование артефактов ПО). Сформулирован базовый принцип предлагаемой методической модели - «расширение возможностей, а не замещение мышления», при котором ИИ ускоряет выполнение рутинных операций, но все этапы анализа, принятия решений и обоснования сохраняются за обучающимся.

Список источников

1. Anahita Golrang, Kshitij Sharma. The role of AI in shaping educational experiences in computer science: A systematic review. - Computers in Human Behavior: Artificial Humans. Volume 6 [Электронный ресурс]: <https://www.sciencedirect.com/science/article/pii/S2949882125000830?via%3Dihub>

2. Said Elnaffar, Farzad Rashidi, Abedallah Zaid Abualkishik. Teaching with AI: A Systematic Review of Chatbots, Generative Tools, and Tutoring Systems in Programming Education. - International Journal of Learning, Teaching and Educational Research Vol. 25, No. 1, pp. 1-28.

3. Georgia Institute of Technology. SCI Reimagines Software Engineering Education Through AI, Sustainability, and Peer Mentorship // ACM CompEd 2025. [Электронный ресурс]:

<https://sci.cc.gatech.edu/external-news/sci-reimagines-software-engineering-education-through-ai-sustainability-and-peer>

4. GenAI-SWE. [Электронный ресурс]: <https://enlight.ugent.be/university-about-us/funded-projects/1583-genai-swe>

5. Зильберман Н.Н. Генеративный ИИ в высшем образовании: условия поддержки метакогнитивной регуляции студентов. Открытое образование. 2026;30(1):15-22. <https://doi.org/10.21686/1818-4243-2026-1-15-22>.

6. Yizhou Fan, Luzhen Tang, Huixiao Le, Kejie Shen, Shufang Tan, Yueying Zhao, Yuan Shen, Xinyu Li, Dragan Gašević. Beware of Metacognitive Laziness: Effects of Generative Artificial Intelligence on Learning Motivation, Processes, and Performance. British Journal of Educational Technology, 56 (2025) 2, S. 489-530

Сведения об авторе

Панюкова Екатерина Владимировна, канд. пед. наук, доцент, доцент кафедры «Информатика и вычислительная техника» ФГБОУ ВО «Самарский государственный технический университет», г. Самара, Россия

Information about the author

Panyukova Ekaterina Vladimirovna, candidate of pedagogical sciences, associate professor of the Department of Computer Science, Federal State Budgetary Educational Institution of Higher Education «Samara State Technical University», Samara, Russia.