

## **ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ПРОЕКТОВ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ РАЗВИТИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ**

**Н.А. Болгов, В.К. Кемайкин, А.Ю. Ключин**

© Болгов Н.А., Кемайкин В.К.,  
Ключин А.Ю., 2026

*Аннотация.* В статье показано, что следующая волна изменений в методологиях разработки программного обеспечения и проектов информационных систем связана с появлением LLM-моделей и ChatGPT. Отмечено, что в ближайшем будущем для профессии «программист» станет не так важно знание языков программирования. Подчеркнуто, что LLM-модели и ChatGPT уже сейчас могут переводить аналитику непосредственно в машинные коды.

*Ключевые слова:* интеллектуальные системы, искусственный интеллект, бизнес-процессы, LLM-модели, ChatGPT.

LLM, или Large Language Model (с англ. «большая языковая модель»), – это языковая модель со множеством параметров. Данная разновидность искусственного интеллекта (ИИ) обучена понимать и генерировать текст любого типа: от технической инструкции до шутки. LLM – это сложные нейросетевые архитектуры, предназначенные для понимания и генерации естественного языка. Они обучаются на огромных объемах текстовых данных и обладают способностью распознавать контекст, извлекать смыслы и отвечать на запросы пользователя естественным языком [1].

Принципы работы LLM:

1) самообучение: модели обучаются методом self-supervised learning, т.е. сеть учится предсказывать пропущенное слово в контексте большого количества предложений;

2) трансформация текста: после обучения модель способна трансформировать исходный текст в требуемый формат, выполняя разнообразные задачи (генерацию ответов, классификацию, парсинг данных и др.);

3) гибкость применения: возможность настройки через prompt engineering (подготовку запроса) позволяет настраивать поведение модели под специфическую задачу.

За последние годы LLM, такие как ChatGPT от OpenAI, Bard от Google, YandexGPT от «Яндекса», GigaChat от «Сбера», стали неотъемлемой частью многих инновационных проектов и решений на рынке, которые предлагают большие компании. Способность понимать естественный язык и генерировать осмысленные тексты делает LLM ценным инструментом для различных областей, таких как анализ текстов, генерация кода и суммирование диалогов. Однако использование LLM сопряжено с рядом ограничений и рисков, включая непредсказуемые ответы и высокие требования к вычислительным ресурсам. В данной работе будут рассмотрены практические аспекты применения LLM в проектировании и разработке приложений, управлении проектами, оптимизации рабочих процессов и поддержании командной работы, а также перспективы развития.

Рассмотрим историю становления больших языковых моделей [2].

1. Период до 2017 года: эра статистических и ранних нейросетевых моделей.

Развитие языкового моделирования началось с использования статистических методов, в частности N-граммных моделей, которые предсказывали следующие слова на основе частотности их появления в тренировочных данных. Эти модели испытывали трудности с обработкой длинных зависимостей и страдали от эффекта «проклятия размерности». С середины 2000-х годов стали появляться первые нейросетевые подходы – нейросетевые N-граммы на основе полносвязных сетей, а затем и рекуррентные нейронные сети. Несмотря на их способность обрабатывать последовательности переменной длины, они имели фундаментальное ограничение – проблему затухающего градиента. Сети забывали или теряли связь со всем предыдущим контекстом, который им ранее задавали, что затрудняло обучение на длинных последовательностях.

2. 2017 год: появление архитектуры Transformer.

Переломным моментом стало появление архитектуры Transformer. Механизм самовнимания (self-attention) позволял модели одновременно анализировать все слова во входной последовательности и выявлять зависимости между ними независимо от длины контекста. Это решало проблему затухающего градиента и позволяло эффективно распараллеливать вычисления. Transformer обеспечил скачок эффективности, эквивалентный примерно двум годам алгоритмического прогресса, и заложил архитектурную основу для всех последующих крупных языковых моделей.

3. 2018–2019 годы: появление первых больших языковых моделей.

На основе архитектуры Transformer были созданы первые действительно большие языковые модели. В 2018 году появились BERT от Google и GPT-1 от OpenAI. BERT использовал двунаправленное обучение и стал прорывом в задачах понимания языка, а GPT-1 продемонстрировала

эффективность авторегрессионного подхода для генерации текста. В 2019 году OpenAI представила GPT-2 с 1,5 млрд параметров, которая показала беспрецедентные возможности генерации связных текстов, что вызвало активные дискуссии об этичности использования таких моделей.

4. 2020 год: эпоха масштабирования и few-shot-обучения.

Выпуск GPT-3 со 175 млрд параметров ознаменовал новый этап. Модель продемонстрировала способность к few-shot и zero-shot learning – решению задач без специального дообучения, лишь на основе нескольких примеров в промпте. Это показало, что простое масштабирование размеров модели и тренировочных данных может приводить к качественному скачку в способностях.

5. 2022 год: демократизация и практическое применение.

Запуск ChatGPT на основе GPT-3.5 сделал мощные языковые модели доступными массовому пользователю. Всего за два месяца сервис набрал 100 млн пользователей. Это доказало практическую ценность больших языковых моделей и стимулировало их интеграцию в различные сервисы и продукты. Началось активное развитие индустрии тонкой настройки и адаптации моделей под конкретные задачи.

6. 2023 год: мультимодальность и эффективность.

Выпуск GPT-4 продемонстрировал, что нейросети способны одновременно обрабатывать текст и изображения, открыл эру мультимодальных моделей. Параллельно получили распространение разреженные архитектуры Mixture of Experts (MoE), такие как Mixtral 8x7B, которые значительно ускоряли вывод за счет активации только части параметров для каждого запроса. Широкое внедрение методов квантизации (QLoRA, GPTQ) позволило эффективно запускать большие модели на потребительском оборудовании.

7. 2024–2025 годы: специализация и оптимизация эффективности.

Современный этап характеризуется смещением фокуса с простого увеличения размеров моделей на оптимизацию их эффективности. Развиваются открытые модели (Llama3, Mistral), создаются специализированные архитектуры для работы с длинным контекстом, совершенствуются методы сжатия. Появляются модели, ориентированные на конкретные домены и задачи и при этом сохраняющие высокую производительность при меньших вычислительных затратах. Активно развиваются методы оценки качества моделей для сложных задач, требующих логического мышления и глубокого понимания контекста.

Покажем примеры проектов, построенных по методологии LLM [3].

Пример 1: Resume Analyzer.

Проект представляет собой систему автоматического анализа резюме, которая помогает рекрутерам быстро оценивать соответствие кандидата позиции. Модель использует методы обработки естественного

языка для выделения ключевых компетенций, образования и опыта работы соискателя.

Преимущества:

быстрая оценка множества резюме;

повышение объективности отбора благодаря формализации критериев соответствия.

Недостатки:

возможны ошибки интерпретации текста, требующие ручной проверки;

требуется настройка модели под конкретный набор вакансий и отраслей.

Пример 2: Code Generator.

Это система автоматической генерации кода на основе описания задачи. Пользователь описывает проблему на естественном языке, а модель создает соответствующий фрагмент кода. Данный инструмент ускоряет разработку прототипов и снижает нагрузку на программистов.

Преимущества:

экономия времени на написании шаблонного кода;

поддержка нескольких языков программирования.

Недостатки:

качество сгенерированного кода зависит от детализации запроса;

необходимость дополнительной проверки и доработки вручную.

Пример 3: Dialog Summarizer.

Модель суммирует длинные технические обсуждения или встречи, предоставляя краткую выжимку ключевых моментов. Эта функция полезна для менеджеров проектов, аналитиков и руководителей, желающих оперативно получать важную информацию.

Преимущества:

значительное сокращение времени на ознакомление с длинными документами;

удобство восприятия информации.

Недостатки:

потеря деталей и нюансов при сжатии текста;

зависимость от качества исходного материала.

Остановимся на преимуществах и ограничениях метода LLM.

Преимущества:

масштабируемость (LLM позволяют работать с большим объемом текстовой информации одновременно);

экономия времени (автоматизация рутинных операций повышает общую скорость разработки);

доступность (локальное развертывание моделей (например, Ollama) дает возможность использовать их без передачи данных третьим лицам).

Ограничения:

непредсказуемость (иногда модели дают некорректные или несвязные ответы («галлюцинации»));

Перспективы развития методологии LLM связаны с несколькими ключевыми тенденциями:

интеграцией с платформами low-code/no-code (позволит менее технически подкованным сотрудникам использовать возможности LLM для создания собственных решений);

улучшенными механизмами контроля качества (будут разработаны методики минимизации риска «галлюцинаций», повышения надежности выводов);

специализацией моделей (появится больше специализированных моделей, настроенных под конкретные отрасли и задачи).

Таким образом, методология LLM продолжает расширяться, предлагая эффективные способы автоматизации и улучшения процессов разработки программного обеспечения.

Рассмотрим пример проектирования и разработки приложений с применением LLM [4].

1. Генерация технического задания (ТЗ) и спецификаций.

Одним из первых этапов любого IT-проекта является создание ТЗ и спецификации. Традиционно этот этап требует значительного количества времени и усилий команды аналитиков и инженеров. Использование LLM позволяет ускорить данный процесс путем автоматического формирования ТЗ на основании описания идеи продукта пользователем. Пример такой системы представлен компанией Sber в проекте GigaChain (рис. 1). Здесь используется LangChain Expression Language (LCEL) для объединения нескольких модулей, позволяющих автоматически формировать ТЗ на основе простых пользовательских входных данных [5].



Рис. 1. Интерфейс платформы Sber GigaChain

На рис. 2 показан интерфейс CoPilot в «Битрикс24». Это ассистент, который работает на технологии ИИ и встроен в ежедневные рабочие сценарии. Он автоматически расшифровывает звонки с клиентами, выделяет ключевые моменты и заполняет карточку сделки в CRM без участия менеджера [6].

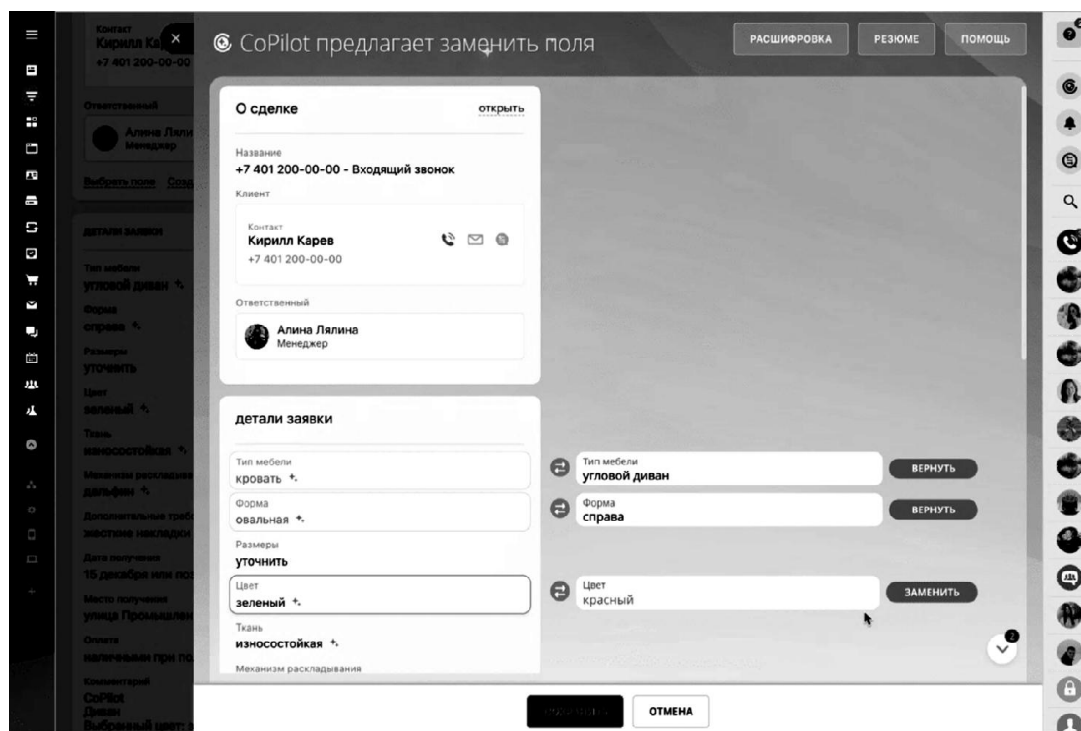


Рис. 2. Интерфейс CoPilot в «Битрикс24»

Решения на основе нейронных сетей также можно применять в целях обеспечения информационной безопасности, т.е. в качестве интеллектуального помощника для получения аналитических данных и рекомендаций. Например, Copilot for Security – это генеративный ИИ-помощник, который повышает эффективность и возможности защитников, работая с машинной скоростью и в широком масштабе. Важно понимать, что это не просто инструмент для аналитиков, которые работают в центре мониторинга безопасности (Security Operations Center). Его применение охватывает множество ролей, включая системных администраторов, аналитиков внутренних рисков и администраторов управления доступом, создает дополнительную ценность для заказчиков. Инструмент может генерировать комплексные отчеты для совета директоров как в технической, так и в нетехнической версиях, что упрощает коммуникацию и помогает принимать обоснованные решения об инвестировании в сферу безопасности.

Решения на основе нейронных сетей применяют, например, и в Cortex XSOAR, который помогает командам безопасности справляться с большим объемом инцидентов. Платформа объединяет данные из

различных инструментов безопасности, автоматизирует рутинные задачи и стандартизирует процессы реагирования.

Рассмотрим управление проектом с поддержкой LLM.

#### 1. Планирование и управление ресурсами.

Компания Analytics Vidhya предлагает идею построения автоматизированного помощника для планирования задач и оценки трудозатрат. На основе введенных данных о проекте (цели, сроков, доступных ресурсов) LLM формирует оптимальный график работ и распределяет задачи среди участников команды [7].

#### 2. Мониторинг прогресса и контроль рисков.

Мониторинг хода выполнения проекта и своевременное выявление потенциальных рисков – это еще одно направление, где активно внедряется технология LLM. При использовании методов Prompt Engineering, таких как zero-shot и few-shot, можно настроить систему для отслеживания изменений в ходе проекта и прогнозирования возможных отклонений от плана. Подобные подходы описаны в исследовании SSL Team [8].

#### 3. Поддержка командной работы и коммуникации чат-ботов.

Чат-бот может отвечать на вопросы сотрудников относительно структуры проекта, статуса задач, доступных ресурсов и рекомендаций по улучшению производительности. Подобный подход применялся в ряде российских компаний, включая «Сбер» и «Яндекс».

#### 4. Помощь в принятии решений.

Команда проекта может задать системе вопрос о целесообразности выбора той или иной технологии или подхода, а LLM предложит аргументированное обоснование своего ответа, отталкиваясь от анализа доступной информации. Эта функция была продемонстрирована в экспериментальной среде TAdviser.

#### 5. Оптимизация рабочего процесса через LLM.

Тестирование программного обеспечения – это трудоемкая процедура, которую можно частично автоматизировать с помощью LLM. Например, система способна проанализировать отчет о проблеме и предложить потенциальные пути ее устранения. Такого рода инструменты были представлены в исследованиях Evidently и показали значительное сокращение времени на тестирование, а также повышение его точности.

#### 6. Улучшение взаимодействия с пользователями.

Чат-боты на основе LLM могут быстро предоставлять пользователям нужную информацию, решать простые проблемы и направлять сложные случаи специалистам. Подобная практика широко используется крупными компаниями, такими как «Сбер» и «Яндекс».

Развитие технологий LLM приведет к тому, что они станут более полезными инструментами для менеджеров проектов и разработчиков [9].

Методология LLM демонстрирует значительный потенциал для повышения эффективности управления проектами и разработки программного обеспечения. Она позволяет автоматизировать многие рутинные операции, улучшает внутренние коммуникации и координацию между участниками проекта, способствует повышению точности прогнозирования рисков и оптимизирует рабочий процесс за счет быстрой генерации ТЗ, прототипов интерфейсов и планов распределения ресурсов [10].

Грамотная интеграция LLM поможет организациям стать конкурентоспособными и достичь новых высот в области цифровой трансформации.

### **Библиографический список**

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. М.: ДМК Пресс, 2017. 654 с.
2. Обзор возможностей ChatGPT-4. URL: <https://habr.com/ru/articles/905596> (дата обращения: 22.09.2025).
3. Dev.to. Примеры приложений на основе больших языковых моделей. URL: [https://dev.to/dasha\\_maliugina/20-examples-of-llm-powered-applications-in-the-real-world-p8c](https://dev.to/dasha_maliugina/20-examples-of-llm-powered-applications-in-the-real-world-p8c) (дата обращения: 22.10.2025).
4. Саттон Р.С., Барто Э.Г. Обучение с подкреплением. М.: Бином, 2017. 402 с.
5. Sber Developers. Large Language Models. URL: <https://developers.sber.ru/help/gigachat-api/large-language-models> (дата обращения: 22.10.2025).
6. Бутл Р. Искусственный интеллект и экономика. Работа, богатство и благополучие в эпоху мыслящих машин. М.: Интеллектуальная Литература, 2022. 432 с.
7. Analytics Vidhya. Projects on LLM. URL: <https://www.analyticsvidhya.com/blog/2023/05/projects-on-llm/> (дата обращения: 22.10.2025).
8. SSL Team. Что такое модели LLM. URL: <https://ssl-team.com/blog/llm-modeli-cto-eto> (дата обращения: 22.10.2025).
9. Зубков А.В. Нейросеть на пальцах: как работает ИИ и как его использовать? М.: Изд. Зубков А.В., 2025. 50 с.
10. Киссинджер Г., Шмидт Э., Хоттенлокер Д. Искусственный разум и новая эра человечества. М.: Альпина PRO, 2022. 200 с.

**DIGITAL TRANSFORMATION OF SOFTWARE  
AND INFORMATION SYSTEMS PROJECTS  
BASED ON THE DEVELOPMENT OF LARGE LANGUAGE MODELS**

**N.A. Bolgov, V.K. Kemaikin, A.Yu. Klyushin**

***Abstract.** The article demonstrates that the next wave of changes in software development methodologies and information systems projects is associated with the emergence of LLM models and ChatGPT. It notes that in the near future, knowledge of programming languages will become less important for the programmer profession. It emphasizes that LLM models and ChatGPT can already translate analytics directly into machine code.*

***Keywords:** intelligent systems, artificial intelligence, business processes, LLM models, ChatGPT.*

Об авторах:

БОЛГОВ Николай Алексеевич – магистрант кафедры информационных систем, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: ko\_bolgov@mail.ru

КЕМАЙКИН Валерий Константинович – кандидат технических наук, профессор кафедры радиолокационных информационных систем, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: vk-kem@mail.ru

КЛЮШИН Александр Юрьевич – кандидат технических наук, доцент кафедры информационных систем, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: klalex@inbox.ru

About the authors:

BOLGOV Nikolay Alekseevich – Master's Student of the Department of Information Systems, Tver State Technical University, Tver. E-mail: ko\_bolgov@mail.ru

KEMAYKIN Valery Konstantinovich – Candidate of Engineering Sciences, Professor of the Department of Information Systems, Tver State Technical University, Tver. E-mail: vk-kem@mail.ru

KLYUSHIN Aleksandr Yuryevich – Candidate of Engineering Sciences, Associate Professor of the Department of Information Systems, Tver State Technical University, Tver. E-mail: klalex@inbox.ru