

Prompt Engineering: Искусство правильного общения с искусственным интеллектом

Введение: Что такое Prompt Engineering?

Prompt Engineering (Промпт-инжиниринг) — это процесс проектирования, оптимизации и тонкой настройки текстовых запросов (пром프트в) для получения максимально точного, релевантного и полезного ответа от систем генеративного искусственного интеллекта (Generative AI).

В отличие от традиционного программирования, где мы пишем жесткие алгоритмы (код), здесь мы общаемся с машиной на естественном языке. Однако ИИ, несмотря на всю свою «сообразительность», не умеет читать мысли. Качество ответа нейросети **напрямую зависит от качества вашего запроса** (принцип "Garbage In, Garbage Out").

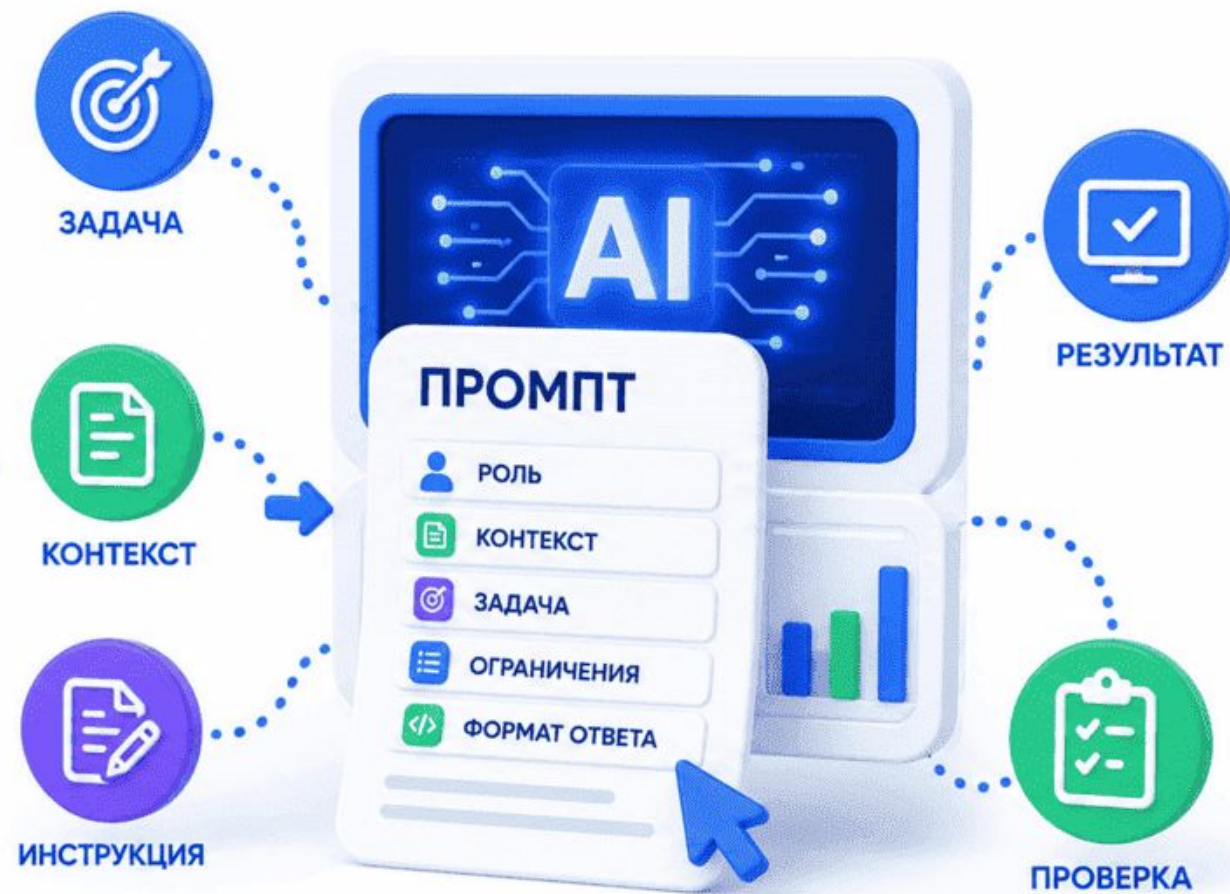
Зачем нужен промпт-инжиниринг

Промпт-инжиниринг делает работу с ИИ точной, предсказуемой и повторяемой. Это особенно важно в бизнес-процессах.

Задача	Как помогает промпт-инжиниринг
Повысить точность	Уточняет цель, контекст и критерии
Получить нужный формат	Задаёт структуру ответа (таблица, JSON, список)
Снизить количество правок	Определяет требования заранее
Сделать результат повторяемым	Использует шаблоны и единые правила
Автоматизировать операцию	Превращает инструкцию в сценарий для AI-агента

Анатомия идеального промпта

Современные исследователи ИИ выделяют несколько ключевых компонентов, которые должны присутствовать в хорошем промпте. Один из популярных фреймворков — **CREATE** или **RTSO** (Role, Task, Steps, Output).



Роль (Role):

Кем должен притвориться ИИ? (Например: *"Действуй как старший финансовый аналитик с 10-летним опытом..."*).

Задача (Task):

Что конкретно нужно сделать? (Например: *"Проанализируй этот отчет и найди три главные причины падения выручки."*).

Контекст (Context):

Какая фоновая информация нужна ИИ для понимания ситуации? (Например: *"Наша компания продает облачное ПО для малого бизнеса. В прошлом квартале мы повысили цены на 15%."*).

Промпт vs промпт-инжиниринг

Промпт — это конкретный запрос к ИИ, а **промпт-инжиниринг** — процесс его проектирования и оптимизации.

Критерий	Промпт	Промпт-инжиниринг
Что это	Инструкция	Методология работы с инструкциями
Цель	Получить результат	Сделать результат качественным и повторяемым
Контекст	Может отсутствовать	Проектируется специально
Тестирование	Не обязательно	Обязательная часть
Применение	Разовый запрос	Повторяемые задачи и процессы

Промпт-инжиниринг: пример и разбор реального сценария

Промпт-инжиниринг на практике — это превращение размытой задачи в конкретную инструкцию. Вот пример промпт-инжиниринга для анализа обращений клиентов.

Базовый запрос: «Проанализируй обращение клиента».

Инженерный подход: «Ты — аналитик поддержки с опытом работы более 15 лет. Проанализируй обращение клиента. Определи тональность (положительная/нейтральная/отрицательная), тему (оплата, доступ, техпроблема, доставка, возврат, другое), срочность (высокая/средняя/низкая) и рекомендуемое действие. Используй только информацию из обращения. Если данных недостаточно, укажи "Не определено". Верни результат в таблице.»

Разница очевидна: во втором случае ИИ получает не просто тему, а алгоритм действий, ограничения и формат.

Базовые техники промптинга

1. Zero-Shot Prompting (Запрос без примеров)

Вы просто просите ИИ выполнить задачу, не давая ему образцов того, как это должно выглядеть.

- **Пример:** *"Определи тональность этого отзыва: 'Новый интерфейс приложения просто ужасен, ничего не понятно'. Ответ (Позитивный/Негативный/Нейтральный):"*

2. Few-Shot Prompting (Запрос с примерами)

Вы даете ИИ несколько примеров входных данных и желаемых ответов. Это критически важно для сложных задач, где ИИ должен понять ваш стиль или логику.

- **Пример:** "Отзыв: 'Доставили быстро, все супер.' -> Тональность: Позитивный. Отзыв: 'Коробка помята.' -> Тональность: Негативный. Отзыв: 'Нормальный телефон за свои деньги.' -> Тональность: " (ИИ сам ответит "Нейтральный")."

Продвинутые стратегии (по материалам последних исследований)

Современная литература по ИИ выделяет ряд техник, заставляющих языковые модели «думать» лучше.

1. Chain of Thought (CoT) — Цепочка рассуждений

Если дать ИИ сложную логическую или математическую задачу напрямую, он может ошибиться. Но если попросить его **рассуждать шаг за шагом**, точность резко возрастает.

- **Магическая фраза:** *"Давай думать шаг за шагом"* (Let's think step by step).
- **Как это работает:** ИИ генерирует промежуточные логические шаги, что позволяет ему самому проверять себя и не терять контекст сложной задачи.

2. Tree of Thoughts (ToT) — Дерево мыслей

Эволюция CoT. Модели предлагается сгенерировать несколько альтернативных путей решения задачи (ветвей), оценить каждый из них, отбросить тупиковые и выбрать лучший. Эта техника используется для сложных стратегических задач.

- **Пример промпта:** *"Предложи 3 разных стратегии выхода на новый рынок. Оцени плюсы и минусы каждой, а затем выбери самую надежную и распиши ее подробно."*

3. Retrieval-Augmented Generation (RAG) в промптах

Использование внешних знаний. Вы не полагаетесь на память ИИ, а сначала предоставляете ему текст (например, статью или документ), а затем просите ответить на вопросы **строго** на основе этого текста. Это решает проблему "галлюцинаций" (выдумывания фактов).

Метод	Что это	Когда применять	Сложность
Zero-shot	Задача без примеров	Простые классификации, перевод, суммаризация	Низкая
Role Prompting	Задание роли и контекста	Экспертный анализ, адаптация ответа под аудиторию	Низкая
Few-shot	1–2 примера правильного ответа	Классификация, нестандартный формат, стилизация	Средняя
Декомпозиция	Разбивка задачи на этапы	Сложные многошаговые задачи	Средняя
Structured Output	Фиксированный формат (JSON, таблица)	Интеграция с CRM, API, бизнес-процессами	Средняя
Chain-of-Thought	Пошаговое рассуждение	Логические и аналитические задачи	Высокая
Self-Critique	Самопроверка результата	Контроль качества, фактчекинг	Высокая
ReAct	Рассуждение + действие	AI-агенты, работа с инструментами	Высокая

Главные ошибки при написании промптов

Недостаток контекста: Выброс задачи без предыстории.

Двусмысленность: Использование расплывчатых терминов (например, *"сделай текст лучше"* — что значит "лучше"? Короче? Смешнее? Формальнее?).

Конфликтующие инструкции: Когда в одном длинном промпте одно правило противоречит другому.

Ожидание магии: ИИ не знает нюансов вашего бизнеса, если вы ему их не рассказали.

Лучшая практика 2026: начинайте с Zero-shot. Если результат нестабилен, добавляйте Role Prompting и Few-shot. Для сложных задач используйте Chain-of-Thought. Для автоматизации — ReAct и Self-Critique.

Рекомендуемая литература

Для более глубокого погружения в тему рекомендуется изучить следующие современные работы:

- **"Prompt Engineering for Generative AI"** (James Weaver, 2023-2024) — базовые принципы и паттерны.
- **"The AI Revolution in Medicine"** (Peter Lee, 2023) — отличные примеры того, как правильные промпты могут спасать жизни (показывает важность экспертного контекста).
- Научные статьи (Papers) от OpenAI и Google Research (например, оригинальная статья по *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*, Jason Wei et al.).

Примеры промпт-инжиниринга: готовые шаблоны для бизнеса

Задача

Классификация обращения

Пример промпта (инженерный подход)

«Ты — аналитик поддержки с опытом работы более 15 лет. Определи категорию обращения: оплата, доступ, техпроблема, доставка, возврат, другое. Используй только текст обращения. Верни результат: категория + краткое обоснование.»

Анализ отзыва

«Ты — маркетолог с опытом работы более 10 лет. Проанализируй отзыв клиента. Определи тональность, основную проблему, срочность (высокая/средняя/низкая). Используй только информацию из отзыва.»

Подготовка поста для соцсетей

«Ты — B2B-маркетолог с опытом работы более 10 лет. Напиши пост для Telegram о CRM. Целевая аудитория — руководители отделов продаж. Структура: проблема → решение → результат → призыв. Объем — 800 знаков. Без клише.»

Извлечение данных из документа

«Ты — аналитик с опытом работы более 10 лет. Извлеки из договора: стороны, дату, срок действия, стоимость. Если значение отсутствует, укажи "Не найдено". Формат: таблица.»

Сравнение двух документов

«Ты — юрист с опытом работы более 20 лет. Сравни два договора аренды. Найди различия по ключевым пунктам: срок, стоимость, ответственность. Верни результат в таблице.»

Промпт-инжиниринг в бизнесе

Промпт-инжиниринг применяется в продажах, маркетинге, клиентской поддержке, HR, документообороте и аналитике. Он превращает ИИ в инструмент повторяемых бизнес-операций.

Область	Типовые задачи
Продажи	Анализ переговоров, подготовка коммерческих предложений, анализ сделок
Маркетинг	Генерация контента, анализ аудитории, A/B-тестирование идей, сбор новостей конкурентов
Поддержка	Классификация обращений, подготовка ответов по базе знаний
HR	Анализ резюме, подготовка вопросов для интервью
Документооборот	Извлечение данных из договоров и счетов
Аналитика	Поиск аномалий, подготовка сводок и отчётов

Промпт-инжиниринг для AI-агентов

AI-агент — это система, которая не только генерирует текст, но и выполняет действия (запросы к API, изменение данных, интеграции). Промпт-инжиниринг для AI-агентов определяет цель, инструменты, правила, ограничения и условия завершения.

Пример промпта для AI-агента:

«Ты — AI-помощник отдела продаж. Цель: находить сделки с истекающими договорами (30 дней). Инструмент: CRM. Правила: используй CRM для данных. Ограничения: не изменяй записи. Результат: таблица с клиентом, датой окончания и действием.»

Как тестировать и улучшать промпты

Тестирование промптов — это обязательная часть промпт-инжиниринга, если инструкция используется в бизнес-процессе. Оценивайте по критериям: точность, полнота, стабильность, соблюдение формата, фактическая корректность.

Алгоритм:

1. Подготовьте тестовый набор из 10–20 случаев (включая пограничные).
2. Прогоните промпт и зафиксируйте результаты.
3. Оцените по метрикам (точность, полнота, ошибки).
4. Внесите изменения в промпт.
5. Повторите тест на том же наборе.
6. Сравните версии и выберите лучшую.

Что изменилось в 2026 году?

С появлением «думающих» (reasoning) моделей многие старые техники перестали работать или стали вредными. Эксперты советуют:

- **Убрать развёрнутый Chain-of-Thought (CoT)** — модели научились рассуждать внутри себя.
- **Сократить Few-shot до 1–2 примеров** — 5–6 примеров теперь перегружают контекст.
- **Убрать эмоциональную Role-Play («ты гениальный»)** — она мешает модели решать задачу.
- **Не использовать жесткие схемы рассуждения** — они конфликтуют с внутренним мышлением модели.

Главный совет 2026 года: формулируйте задачу чётко и не мешайте модели думать.

Заключение

Prompt Engineering — это не программирование, это психология общения с другой (искусственной) когнитивной системой. Умение четко формулировать свои мысли, давать нужный контекст и направлять ход рассуждений ИИ — один из самых важных навыков XXI века.

1. Что такое промпт-инжиниринг простыми словами?

Промпт-инжиниринг — это навык правильно ставить задачи нейросети, чтобы получать точный и полезный результат.

Простыми словами, это умение разговаривать с искусственным интеллектом на его языке: вы объясняете задачу максимально ясно (с ролью, контекстом и форматом), а он выполняет её так, как нужно именно вам. Нейросети не читают мысли и не понимают контекст, очевидный для человека — они реагируют только на то, что написано в запросе.

2. Чем промпт-инжиниринг отличается от обычного запроса к ИИ?

Обычный запрос — это вопрос или команда, а **пром프트-инжиниринг** — это системный процесс проектирования **инструкции**. Простой запрос («Напиши статью») даёт неопределённый результат. Промпт-инжиниринг превращает его в управляемую инструкцию: вы добавляете роль, контекст, ограничения, формат и примеры (few-shot). В результате вы получаете не случайный ответ, а стабильный, предсказуемый и повторяемый результат, который можно встроить в бизнес-процесс.

3. Какие методы промпт-инжиниринга самые эффективные?

Самые эффективные методы промпт-инжиниринга зависят от задачи. Для простых задач достаточно **Zero-shot** (чёткая инструкция без примеров). Для классификации и стилизации используйте **Few-shot** (2–5 примеров в промпте). Для логических и аналитических задач — **Chain-of-Thought** (пошаговое рассуждение). Для строгого формата — **Structured Output** (JSON, таблица). Для AI-агентов — **ReAct** (рассуждение + действие). Лучшая практика: начинайте с простого и добавляйте техники только при необходимости.

4. Что такое Few-shot prompting?

Few-shot prompting — это метод промпт-инжиниринга, при котором в промпт добавляют 2–5 примеров правильных **ответов**. Модель использует эти примеры как образец, чтобы понять нужный формат, стиль или принцип классификации. Например, вы показываете модели три примера категоризации обращений, а затем просите классифицировать новое. Few-shot особенно полезен, когда правило сложно описать словами, но легко показать на примерах.

5. Что такое Chain-of-Thought и когда его использовать?

Chain-of-Thought (CoT) — это метод промпт-инжиниринга, при котором модель сначала показывает логику, а потом **даёт ответ**. Он даёт прирост качества на 20–40% для сложных задач: математики, аналитики, планирования, сравнения вариантов. **Важно:** CoT эффективен только на больших моделях (≈ 100 млрд параметров и выше); на малых моделях эффект может быть незначительным или даже отрицательным. Для простых задач CoT не нужен — он только усложнит инструкцию.

6. Нужно ли уметь программировать для промпт-инжиниринга?

Для базового уровня программирование не требуется. Промпт-инжиниринг — это в первую очередь навык формулировки задач, работы с контекстом и ограничениями. Однако технические знания становятся полезными при работе с API, AI-агентами, автоматизацией, RAG, оценкой качества (evals) и интеграцией ИИ в корпоративные системы. Для профессионального роста в этой области программирование — это преимущество, но не обязательное условие на старте.

7. Какие ошибки чаще всего допускают новички в промпт-инжиниринге?

Самые частые ошибки в промпт-инжиниринге: слишком общий запрос, отсутствие роли, недостаток контекста, нет формата, противоречивые требования и перегрузка техниками. Новички часто пишут «сделай хорошо» вместо конкретной инструкции, забывают указать, в каком формате нужен ответ, или смешивают несколько задач в одном промпте. Исправление: всегда формулируйте конкретную задачу, добавляйте роль, контекст, ограничения и формат, проверяйте результат на тестовых данных и улучшайте итерационно.

8. Когда промпт-инжиниринг не нужен?

Промпт-инжиниринг не нужен для простых задач, где достаточно чёткого запроса: перевод коротких фраз, простые вопросы, генерация идей. Если задача решается одним-двумя предложениями без дополнительных уточнений — сложные техники только усложнят процесс. Однако для бизнес-процессов, автоматизации, работы с документами, аналитики и AI-агентов промпт-инжиниринг становится обязательным инструментом.