

POSTNIKOV.AI — SYSTEMS

CONTEXT ENGINEERING KIT

PLAYBOOK #1

Как разговаривать с LLM

От нуля до рабочей системы за один вечер

MAX POSTNIKOV

`postnikov.ai`

Содержание

Context Engineering Kit – Playbook 1	4
Как разговаривать с LLM	4
Глава 1. Почему AI тебя не понимает	6
Проблема	6
Формула	6
Что такое контекст	7
Пример	7
Антисовет	7
Глава 2. 4 уровня контекста	9
Уровень 0: Чужой	9
Уровень 1: Знакомый	9
Уровень 2: Коллега	9
Уровень 3: Лучший друг	10
Сводная таблица	11
Где ты сейчас?	12
Глава 3. Твой контекстный файл	13
Что это	13
Мой опыт	13
Структура	13
Где использовать	15
Сколько это занимает	15
Life-Hack: пусть AI напишет за тебя	15
Глава 4. 10 паттернов, которые работают	17
Часть 1: Фундамент (начни с этих трёх)	17
Паттерн 1: Role + Context + Task	17
Паттерн 2: Chain of Thought	18
Паттерн 3: Few-Shot Examples	19
Часть 2: Справочник (возвращайся по мере необходимости)	20
Паттерн 4: Persona Calibration	20
Паттерн 5: Structured Output	20
Паттерн 6: Iterative Refinement	21

Паттерн 7: Constraint Setting	21
Паттерн 8: Devil's Advocate	22
Паттерн 9: Knowledge Boundary	23
Паттерн 10: Multi-Step Pipeline	24
Шпаргалка: какой паттерн когда	25
Глава 5. Prompt Debugging — когда AI несёт чушь	26
Чеклист: 7 причин плохого ответа	26
Быстрая диагностика	28
Антисовет про email	28
Глава 6. Практика — прокачайся за один вечер	30
Упражнение 1: Паттерн-рефакторинг	30
Упражнение 2: Prompt Debugging в бою	31
Бонус 1. Context Drift — почему AI тупеет в длинных чатах	32
Что происходит	32
Context Rot	32
Compression — встроенная защита	33
4 правила работы с длинными чатами	33
Когда начинать новый чат	34
Бонус 2. Projects — AI, который не забывает	35
Что такое Projects	35
Зачем это нужно	35
Как устроено	36
5 проектов, с которых стоит начать	37
Антисовет	37
Projects vs контекстный файл	37

Context Engineering Kit — Playbook 1

> Как разговаривать с LLM

От нуля до рабочей системы за один вечер

Автор: Max Postnikov Сайт: postnikov.ai

Дисклеймер

Prompt engineering недостаточен.

Я год работаю с AI каждый день — пишу тексты, строю продукты, управляю сообществом из 100+ профессионалов. И вот что я понял: *промт — это 10% успеха. Остальные 90% — контекст.* Умение формулировать промты важно, но без контекста даже идеальный промт даёт средний результат. Этот гайд — про оба.

Этот гайд не про «напиши волшебное слово и AI выдаст шедевр». Таких гайдов в интернете тысячи. Они не работают. Ты, скорее всего, уже пробовал.

Этот гайд про систему. Прочитал — внедрил — работает. Без магии. С понятным пошаговым планом реализации.

Это первый плейбук из серии. Он про фундамент: контекст, паттерны, debug. Следующие выходят на postnikov.ai.

Для кого

Ты уже пользуешься AI. ChatGPT, Claude, Gemini — неважно. Но результат тебя бесит.

AI отвечает:

- > Слишком общо — как Википедия, только хуже
- > Не в тему — ты про бизнес, он про «рассмотрим все стороны вопроса»
- > Длинно и ни о чём — три абзаца воды вместо одного ответа

Знакомо? Тогда ты в правильном месте.

В интернете сотни бесплатных гайдов по промптам. OpenAI и Anthropic публикуют свои. Разница: они учат пользоваться *их* инструментом. Этот плейбук учит пользоваться *любым* AI — и строит систему, которая работает независимо от того, какая модель сейчас лучшая.

Что внутри

1. Почему AI тебя не понимает (спойлер: дело не в модели и не в промпте)
2. 4 уровня контекста — от «чужой» до «лучший друг»
3. Как создать свой контекстный файл за 10 минут
4. 10 паттернов, которые работают — готовые формулы
5. Что делать, когда AI несёт чушь — чеклист debug'a
6. Context drift — почему AI тупеет в длинных чатах (и как с этим жить)
7. Projects — как сделать AI-помощника, который не забывает
8. 2 упражнения — прокачаться за один вечер

В комплекте — готовые инструменты:

- > 5 шаблонов контекстных файлов — открыл, заполнил, AI тебя понимает
- > AI Readiness Scorecard — определи свой уровень за 2 минуты
- > 30-Minute Setup Checklist — настрой систему за полчаса
- > Шпаргалка паттернов — одна страница, все формулы
- > Мета-промпт — AI сам напишет тебе контекстный файл
- > Debug-промпт — одна команда, которая чинит 80% плохих ответов

Как читать

30-40 минут от начала до конца. Но лучше — по одной главе, с практикой. Каждая глава заканчивается действием.

В большинстве глав — блок «Что происходит прямо сейчас»: свежие тренды, о которых стоит знать.

Поехали.

Глава 1. Почему AI тебя не понимает

Я попросил AI помочь мне с текстами для блога. Получил идеально гладкий, абсолютно мёртвый текст. Как будто его написал робот из 2019 года, который прочитал все мотивационные посты LinkedIn и решил, что так люди разговаривают.

Потом я заставил AI прочитать мой блог. Понять мой стиль. Запомнить, как я строю предложения, какие слова использую, где ставлю точки.

Тот же AI. Та же модель. Результат — небо и земля.

Разница? *Контекст.*

> Проблема

Представь: ты заходишь в кабинет к незнакомому консультанту. Садись. И говоришь:

«Напиши мне стратегию.»

Он начинает говорить. Уверенно, гладко, долго. Про SWOT-анализ, про сегментацию, про конкурентные преимущества. Звучит правильно. Но это *не про тебя*. Это generic-стратегия, которую он мог бы дать кому угодно.

Это ровно то, что ты делаешь с AI каждый день.

> Формула

Качество ответа = f(Качество контекста)

Не «качество промпта». *Контекста*. Промпт — это часть контекста, но не весь контекст.

> Что такое контекст

Контекст — это всё, что AI знает о тебе и о твоей проблеме, когда начинает отвечать:

- > Кто ты — роль, опыт, уровень
- > Что ты делаешь — проект, задача, дедлайн
- > Зачем — цель, критерии успеха
- > Как ты хочешь получить ответ — формат, длина, стиль
- > Что уже пробовал — чтобы не повторять

Без этого AI вынужден *угадывать*. И угадывает в сторону среднего. Среднего уровня, среднего запроса, средней аудитории. Поэтому ответ и звучит как Википедия.

> Пример

Без контекста:

«Напиши пост для LinkedIn.»

AI напишет generic-мотивашку с эмодзи. Потому что большинство людей просят именно это.

С контекстом:

«Я — технический предприниматель, пишу для аудитории 30-45 лет, IT-профессионалов. Тон: прямой, без булшита, с личным опытом. Тема: почему я перестал нанимать джуниоров и заменил их AI-агентами. Формат: 150-200 слов, hook в первой строке.»

Тот же AI. Та же модель. Но ответ будет в 10 раз точнее, потому что ты дал контекст.

> Антисовет

Вот что все рекомендуют: «*Формулируй промпт максимально подробно.*»

Вот почему это не работает: подробный промпт без контекста — это длинное письмо незнакомцу. Ты можешь написать три абзаца инструкций, но если AI не знает *кто ты* — он всё равно выдаст средний результат. Только длиннее.

Не надо подбирать слова. Надо *дать контекст*.

→ *Действие*: Открой свой последний диалог с AI. Посмотри на свой первый промпт. Сколько из пяти элементов контекста (кто, что, зачем, как, что уже пробовал) ты дал? Скорее всего — ноль или один.

Что происходит прямо сейчас

Prompt Engineering → *Context Engineering* → *Harness Engineering*. В марте 2026 индустрия заговорила о новом термине: *Harness Engineering*. Идея: модель — это двигатель, но без обвязки (*harness*) из ретраев, валидации, *guard rails* и управления контекстом — двигатель бесполезен. *Prompt engineering* был про слова. *Context engineering* — про информацию. *Harness engineering* — про *систему вокруг AI*.

Чтобы начать разбираться в *Harness Engineering* тебе нужно хорошо понять обе предыдущие ступени. Поэтому ты здесь. Поэтому давай читать дальше.

Глава 2. 4 уровня контекста

Не весь контекст одинаково полезен. Есть иерархия — от «чужой» до «лучший друг». Чем выше уровень, тем лучше AI тебя понимает.

> Уровень 0: Чужой

Ты: "Помоги мне с текстом для блога."
AI: *5 абзацев мотивационной воды с эмодзи
и фразой "В современном мире..."*

Ноль контекста. AI не знает кто ты, о чём пишешь, для кого, в каком стиле. Угадывает среднее. А среднее в интернете — это LinkedIn-мотивашка.

Результат: 90% людей работают на этом уровне и жалуются, что «AI бесполезен». Нет, бро. Бесполезен не AI. Бесполезен ноль контекста.

> Уровень 1: Знакомый

Ты: "Я — маркетолог в B2B SaaS. Напиши cold email
потенциальному клиенту — CEO стартапа на 50 человек.
Продукт: платформа аналитики. Тон: профессиональный,
без агрессии. 3-4 предложения."

Ты дал: роль, аудиторию, тон, формат. AI понимает *задачу*.

Результат: Ответ уже на 70% ближе к тому, что нужно. Но каждый раз нужно заново объяснять кто ты. Каждый. Раз.

> Уровень 2: Коллега

На этом уровне AI знает тебя *до начала разговора*. Ты не объясняешь кто ты и что делаешь — это уже записано.

Как? Через *системный контекст*: контекстный файл, system prompt, или инструкции проекта.

```
# Системный контекст
Я — Алексей, маркетолог в B2B SaaS-стартапе (20 человек).
Пишу на русском. Тон: прямой, без воды.
Аудитория: CTO и VP Engineering, 30-45 лет.
Мой продукт: платформа аналитики для инженерных команд.
```

Теперь:

```
Ты: "Напиши cold email для CTO."
AI: *персонализированный email, правильный тон, правильный продукт*
```

Результат: AI работает как коллега, который знает контекст. Не надо повторять кто ты.

> Уровень 3: Лучший друг

Вот тут начинается магия.

Я задал Claude простой вопрос: «Расскажи про этот сервис.» Ожидал стандартный обзор. А Claude не просто рассказал про сервис — он добавил, в каких *конкретно моих проектах* этот сервис может быть полезен. С примерами. Без подсказки.

Это уровень 3. AI знает не только кто ты и что делаешь, но и:

- > Твой стиль — как ты пишешь, какие слова используешь
- > Твои предпочтения — что тебе нравится и что бесит
- > Твою историю — что работало, что нет
- > Твои проекты — и как они связаны между собой

```
# Расширенный контекст
## Кто я
Алексей, маркетолог. B2B SaaS, аналитика для инженеров.

## Мой стиль
- Короткие предложения. Без воды.
- Конкретные цифры вместо "значительный рост"
- Никаких эмодзи в рабочей переписке
- Юмор допустим, но сухой

## Что работает
- Cold emails с вопросом в теме (open rate 34%)
- Кейсы с конкретными метриками

## Что НЕ работает
- "Надеюсь, у вас всё хорошо" — удаляй без чтения
- Длинные emails > 5 предложений
- Generic value propositions
```

Результат: AI пишет как ты. Не «за тебя» — а именно так, как ты бы написал, только быстрее.

> Сводная таблица

Уровень	Название	Что знает AI	Твои усилия	Качество
0	Чужой	Ничего	Минимум	10%
1	Знакомый	Задачу	Каждый промпт	60%
2	Коллега	Тебя + задачу	Один раз настроить	85%
3	Лучший друг	Тебя + стиль + историю	Один раз + обновлять	95%

> Где ты сейчас?

Большинство людей застряли на уровне 0-1. Перескочить на уровень 2 можно за 10 минут — об этом следующая глава.

→ *Действие*: Определи свой текущий уровень. Если ты на 0-1, следующая глава покажет, как перейти на 2 за 10 минут.

Что происходит прямо сейчас

Context Rot. Качество ответов AI деградирует задолго до того, как контекстное окно заполнится полностью. Даже если модель поддерживает 1 миллион токенов, это не значит, что она одинаково хорошо работает на всём протяжении. Context rot — реальная проблема, и в главе 7 мы разберём, как с ней бороться.

Глава 3. Твой контекстный файл

> Что это

Контекстный файл — это текст, который AI читает *до начала разговора с тобой*. Как записка для нового коллеги: «Вот кто я, вот что мы делаем, вот как здесь принято.»

В Claude Code этот файл называется CLAUDE.md. В ChatGPT — Custom Instructions. В Gemini — System Instructions. Название разное, суть одна: *напиши один раз — AI понимает тебя каждый раз.*

> Мой опыт

Мой собственный контекстный файл прошёл 10+ итераций. Я дорабатываю его примерно раз в месяц.

Почему? Потому что человек меняется. Его проекты меняются. Его приоритеты меняются. *Твой контекстный файл должен меняться вместе с тобой.*

Первая версия была три строчки: кто я, что делаю, на каком языке отвечать. Текущая — полноценный документ с моим стилем, моими проектами, моими предпочтениями и антипредпочтениями.

И знаешь что? Разница между первой и текущей версией — как между уровнем 1 и уровнем 3 из предыдущей главы. Тот же AI, но *другой разговор.*

> Структура

Хороший контекстный файл содержит 4-5 блоков:

1. Кто я (2-3 строки)

```
## Кто я
Мария, UX-дизайнер. Работаю в финтех-стартапе (30 чел).
5 лет опыта, специализация — мобильные интерфейсы.
```

Зачем: AI калибрует уровень. Не будет объяснять что такое wireframe.

2. Что я делаю (2-3 строки)

```
## Текущий проект
Редизайн мобильного приложения для торговли акциями.
Целевая аудитория: 25-35, новички в инвестициях.
Дедлайн: июнь 2026.
```

Зачем: AI понимает контекст задач. Не предложит Enterprise-решения для consumer-продукта.

3. Как отвечать (3-5 правил)

```
## Как мне помогать
- Отвечай конкретно: wireframe > описание
- Если не уверен — спроси, не додумывай
- Критикуй мои решения, если видишь проблему
- Формат: короткие пункты, не эссе
- Язык: русский, технические термины на английском
```

Зачем: AI подстраивается под твой стиль работы.

4. Что НЕ делать (это важнее, чем кажется)

```
## Чего НЕ делать
- Не предлагай Figma-плагины (я использую Sketch)
- Не давай общих советов типа "проведите A/B тест"
- Не начинай ответ с "Отличный вопрос!"
```

Зачем: AI перестаёт предлагать то, что тебя бесит. Этот блок — самый недооценённый.

5. Примеры моей работы (уровень 3)

```
## Мой стиль
Пример хорошего описания экрана:
"Экран онбординга: 3 шага. Шаг 1 — имя и email.
Шаг 2 — выбор стратегии (conservative/balanced/aggressive).
Шаг 3 — первый депозит. Progress bar сверху."
```

Зачем: AI копирует твой стиль формулировок, а не генерирует свой.

> Где использовать

AI-инструмент	Куда положить контекст
Claude (claude.ai)	Project Knowledge или начало чата
ChatGPT	Custom Instructions → «What would you like ChatGPT to know about you?»
Gemini	System Instructions в Google AI Studio
Claude Code	Файл <code>CLAUDE.md</code> в корне проекта
Любой чат	Скопируй текст в начало нового чата

> Сколько это занимает

Первый раз: 10 минут. Заполни 4 блока (кто, что, как, не надо). Этого хватит для уровня 2.

Развить до уровня 3: ещё 10 минут. Добавь примеры работы и предпочтения.

Обновлять: 2 минуты раз в месяц (или когда меняется проект).

> Life-Hack: пусть AI напишет за тебя

Не хочешь писать контекстный файл сам? Пусть AI задаст тебе вопросы и соберёт его из ответов.

Вот промпт — скопируй и вставь в любой чат:

Ты — эксперт по настройке AI-ассистентов.

Твоя задача: помочь мне создать мой персональный контекстный файл (system prompt / CLAUDE.md / Custom Instructions).

Задавай мне вопросы ПО ОДНОМУ. Жди ответа перед следующим вопросом.

Начни с самых важных:

1. Кто я и чем занимаюсь
2. Над чем работаю сейчас
3. Как я предпочитаю получать ответы
4. Что меня бесит в ответах AI
5. Примеры моего стиля (если применимо)

Не ограничивайся только этими вопросами. Сам реши, о чем еще меня спросить исходя из моих ответов на другие вопросы.

Наша с тобой цель — собрать идеальный системный промпт для нашей совместной работы.

После всех вопросов — собери готовый контекстный файл в формате markdown. Покажи мне и спроси, что поменять.

Да, это мета: AI пишет инструкцию для самого себя. Но это работает. И это в 3 раза быстрее, чем писать с нуля.

→ *Действие*: Открой любой AI-чат. Вставь промпт выше. Ответь на 5 вопросов. Получи свой контекстный файл за 5 минут. Или, если хочешь контроль — заполни шаблон из приложения вручную.

Что происходит прямо сейчас

Контекстный файл растёт — и это проблема. Люди начинают с 10 строчек, а через месяц у них 3000 слов. AI тонет в инструкциях и начинает игнорировать половину.

3 способа держать файл коротким:

1. *Выноси детали в Projects*. Контекстный файл = кто ты и как работать. Детали конкретных задач — в Project Knowledge (глава 8). Файл остаётся на 300-500 слов.
2. *Правило «одного экрана»*. Если файл не помещается на один экран — он слишком длинный. Убери всё, что AI может понять из контекста задачи.
3. *Ревизия раз в месяц*. Удаляй правила, которые AI уже и так выполняет. Если ты месяц не напоминал «не используй эмодзи» — значит, AI это усвоил. Удаляй.

Глава 4. 10 паттернов, которые работают

Паттерн — это проверенная формула. Не промпт для копирования, а структура, которую ты заполняешь своим контекстом.

Каждый паттерн: название → когда использовать → формула → пример → частая ошибка.

Антисовет: Не пытайся запомнить все 10. Паттерны 1-3 — это фундамент, они покрывают 80% задач. Освой их. Паттерны 4-10 — справочник. Возвращайся к ним, когда появится конкретная задача.

> Часть 1: Фундамент (начни с этих трёх)

> Паттерн 1: Role + Context + Task

Когда: Подходит для 80% задач. Твой «дефолтный» режим.

Формула:

```
Ты — [роль].
Контекст: [ситуация].
Задача: [что сделать].
Формат: [как выдать результат].
```

Пример:

Ты — опытный продуктовый маркетолог.
Контекст: Я запускаю SaaS для управления задачами.
Целевая аудитория — тимлиды в IT-компаниях 50-200 человек.
Задача: Напиши 3 варианта заголовка для landing page.
Формат: Каждый вариант — одна строка + пояснение в скобках.

Частая ошибка: «Ты — лучший в мире эксперт по всему.» Чем конкретнее роль, тем лучше ответ. «Продуктовый маркетолог в B2B SaaS» > «маркетолог» > «эксперт».

> Паттерн 2: Chain of Thought

Когда: Нужно рассуждение, а не прямой ответ. Анализ, стратегия, решение проблемы.

Формула:

[Задача].
Думай шаг за шагом:
1. Сначала определи [X].
2. Затем проанализируй [Y].
3. На основе этого предложи [Z].
Покажи свои рассуждения.
Задай уточняющие вопросы.

Пример:

Мой конкурент снизил цены на 30%.
Думай шаг за шагом:
1. Почему он мог это сделать? (3 гипотезы)
2. Как это повлияет на наш рынок?
3. Какие у меня варианты ответа?
Покажи рассуждения, потом — рекомендацию.
Задай уточняющие вопросы, если нужно.

Частая ошибка: Использовать Chain of Thought для простых задач. «Думай шаг за шагом: напиши tweet» — это overkill.

> Паттерн 3: Few-Shot Examples

Когда: AI не понимает формат, который тебе нужен. Проще показать, чем объяснять.

Формула:

Задача: [описание].

Примеры:

Вход: [пример 1 вход]

Выход: [пример 1 выход]

Вход: [пример 2 вход]

Выход: [пример 2 выход]

Теперь:

Вход: [твой запрос]

Выход:

Пример:

Перепиши заголовки в стиле нашего блога.

Примеры:

Вход: "Как использовать машинное обучение в маркетинге"

Выход: "ML в маркетинге: 3 кейса, которые работают"

Вход: "Обзор лучших инструментов для аналитики данных"

Выход: "5 аналитических инструментов, без которых мы не выживем"

Теперь:

Вход: "Преимущества автоматизации тестирования ПО"

Выход:

Частая ошибка: Один пример — мало. AI может «зацепиться» за случайную деталь. Два-три примера дают паттерн.

> Часть 2: Справочник (возвращайся по мере необходимости)

> Паттерн 4: Persona Calibration

Когда: Нужен конкретный стиль или тон. Особенно полезно для контента.

Формула:

Пиши в стиле [описание персоны].

Характеристики стиля:

- [черта 1]
- [черта 2]
- [черта 3]

НЕ делай: [антипаттерны].

Пример:

Пиши как опытный технический блогер, который:

- Объясняет сложное простыми словами
- Использует аналогии из повседневной жизни
- Не боится сказать "я не знаю" или "это хайп"
- Пишет на русском с английскими техническими терминами

НЕ делай:

- Не используй эмодзи
- Не начинай с "В современном мире..."
- Не будь снисходительным

Частая ошибка: «Пиши как Стив Джобс» — слишком абстрактно. Описывай черты стиля, а не имя. AI не знает, как именно ты представляешь себе «Стива Джобса».

> Паттерн 5: Structured Output

Когда: Нужен JSON, таблица, список, или любой структурированный формат.

Формула:

[Задача].
Выдай результат в формате:
[пример структуры]

Пример:

Проанализируй 3 конкурента.
Выдай результат в таблице:

Конкурент	Цена	Сильные стороны	Слабые стороны	Наше преимущество

Частая ошибка: Не показывать пример структуры. «Выдай в виде таблицы» — AI угадывает колонки. Покажи какие колонки нужны.

> Паттерн 6: Iterative Refinement

Когда: Результат нужно улучшать поэтапно. Тексты, дизайн, стратегия.

Формула:

Шаг 1: Сделай [черновик/первую версию].
Шаг 2: Критически оцени результат по критериям: [критерии].
Шаг 3: Улучши, исправив найденные проблемы.
Покажи финальную версию и что именно изменил.

Частая ошибка: Просить «напиши идеально с первого раза». Два прохода с критикой лучше одного «идеального».

> Паттерн 7: Constraint Setting

Когда: AI слишком многословен, уходит не туда, или добавляет ненужное.

Формула:

[Задача].

Ограничения:

- Максимум [N] слов/предложений/пунктов
- Только [тема/область]
- Без [чего не надо]
- Уровень: [для кого]

Пример:

Объясни что такое Kubernetes.

Ограничения:

- Максимум 5 предложений
- Для менеджера, не разработчика
- Без технических деталей про pods и nodes
- Одна аналогия из реальной жизни

Частая ошибка: Ограничивать только длину. «Напиши коротко» — AI не знает, что для тебя «коротко». Дай число.

> Паттерн 8: Devil's Advocate

Когда: Нужна критика, а не поддакивание. Проверка идеи, плана, решения.

Формула:

Я планирую [решение/идею].

Выступи как жёсткий критик:

1. Найди 3-5 слабых мест
2. Для каждого: почему это проблема + конкретный сценарий провала
3. Предложи как закрыть каждое слабое место

Не будь вежливым — будь полезным.

Пример:

Я планирую уйти из найма и стать фрилансером.
Текущая ситуация: зарплата €5000/мес, накоплений на 6 месяцев,
2 потенциальных клиента.

Выступи как жёсткий критик:

1. Найди 3-5 слабых мест в моём плане
2. Конкретные сценарии провала
3. Как закрыть каждый риск

Не будь вежливым.
Ругайся матом, если нужно.

Частая ошибка: Просить «дай обратную связь». AI по умолчанию вежлив и поддерживающ. Явно проси критику. Чем жёстче формулировка — тем честнее ответ.

> Паттерн 9: Knowledge Boundary

Когда: Важно, чтобы AI честно сказал «не знаю», а не выдумывал.

Формула:

[Задача].
Важно:

- Если не уверен — скажи прямо
- Не выдумывай факты и цифры
- Отличай точные данные от предположений
- Помечай степень уверенности:  точно /  вероятно /  не уверен
- Приложи ссылки на источники

Пример:

Какой размер рынка AI-инструментов для HR в Европе?
Важно:

- Только проверяемые данные
- Если точных цифр нет — скажи "не знаю" и предложи где найти
- Помечай:  точно /  оценка /  предположение
- Добавь в конце список источников и ссылки на них

Частая ошибка: Верить AI безоговорочно. Даже с этим паттерном — проверяй ключевые факты. AI галлюцинирует. Это не баг — это свойство технологии.

> Паттерн 10: Multi-Step Pipeline

Когда: Задача сложная, нужно разбить на шаги. Каждый шаг — отдельный вход.

Формула:

```
Мы будем работать в 3 шага. Выполняй по одному.  
Шаг 1: [задача]. Покажи результат и жди подтверждения.  
Шаг 2: [задача на основе шага 1].  
Шаг 3: [финализация].  
Начни с шага 1.
```

Пример:

```
Мы напомним статью в 3 шага:  
Шаг 1: Создай outline — 5-7 пунктов. Покажи и жди моего ОК  
или моих комментов.  
Шаг 2: Напиши первый драфт по утверждённому outline.  
Шаг 3: Отредактируй: убери воду, добавь примеры, сократи до 800 слов.  
Начни с шага 1.
```

Частая ошибка: Давать все шаги сразу и ждать финальный результат.

Лучше: шаг → проверка → следующий шаг.

> Шпаргалка: какой паттерн когда

Задача	Паттерн
Любая стандартная задача	1. Role + Context + Task
Анализ, стратегия, «подумай»	2. Chain of Thought
AI не даёт нужный формат	3. Few-Shot Examples
Нужен конкретный тон/стиль	4. Persona Calibration
Таблица, JSON, структура	5. Structured Output
Текст нужно улучшить	6. Iterative Refinement
Слишком длинно/не в тему	7. Constraint Setting
Проверить идею	8. Devil’s Advocate
Важна точность фактов	9. Knowledge Boundary
Сложная задача, много шагов	10. Multi-Step Pipeline

→ *Действие:* Выбери одну задачу, которую ты делаешь регулярно. примени к ней подходящий паттерн. Сравни результат с тем, что получал раньше.

Что происходит прямо сейчас

Популярные промпт-советы часто не работают. Это не удивительно: советы копируются из блога в блог без проверки, обрастают мифами, устаревают за месяц. Вывод: не копируй чужие промпты. Разберись в паттернах — и пиши свои. Именно поэтому этот гайд учит *структурам*, а не даёт готовые тексты для *copy-paste*.

Глава 5. Prompt Debugging — когда AI несёт чушь

AI ответил плохо. Что делать?

Большинство людей: удаляют чат, открывают новый, пробуют другую формулировку. Повторяют 5 раз. Сдаются.

Есть способ лучше. Prompt debugging — это *системный подход* к тому, чтобы понять, почему AI ответил не так, и исправить это за один шаг.

> Чеклист: 7 причин плохого ответа

Когда AI «не понимает» — пройди по списку сверху вниз. Обычно проблема в первых трёх пунктах.

1. Нет контекста

Симптомы: Ответ generic, «как из учебника», не учитывает твою ситуацию.

Исправление: Добавь кто ты, что делаешь и зачем. Используй паттерн «Role + Context + Task».

Было: «Как улучшить продажи?» *Стало:* «Я — основатель B2B SaaS для HR (ARR \$200K, 30 клиентов). Как увеличить конверсию из trial в paid? Сейчас 8%, хочу 15%. Основная причина отвала — пользователи не завершают onboarding.»

2. Много задач в одном промпте

Симптомы: AI отвечает поверхностно на всё. Или отвечает только на часть.

Исправление: Одна задача — один промпт. Если задач несколько — используй Multi-Step Pipeline.

Было: «Проанализируй конкурентов, напиши стратегию, и создай контент-план на квартал.» *Стало:* «Проанализируй 3 главных конкурента в формате таблицы: цена, ЦА, сильные стороны, слабые стороны.»

3. Нет примеров

Симптомы: AI даёт правильный по смыслу, но неправильный по формату ответ.

Исправление: Покажи 1-2 примера того, что хочешь получить. Используй Few-Shot Examples.

4. Неправильная модель

Симптомы: Ответ слишком простой для сложной задачи. Или слишком медленный для простой.

Исправление:

- > Простые задачи (перевод, форматирование, рутина) → быстрая модель (Haiku, GPT-4o mini)
- > Сложные задачи (анализ, стратегия, код) → сильная модель (Opus, o1)
- > Творческие задачи (тексты, идеи) → средняя модель (Sonnet, GPT-4o)

Если подписка позволяет, я всегда рекомендую использовать самую мощную модель с режимом Extended Thinking. Но помни — этот подход может быстро исчерпать твои лимиты.

5. Слишком длинный промпт

Симптомы: AI «теряется» — отвечает на часть, игнорирует другую.

Исправление: Сократи. Убери всё, без чего можно обойтись. Фокус > объём. Если контекста много — используй контекстный файл (глава 3), чтобы не повторять базу каждый раз.

6. AI «залип» в контексте

Симптомы: В длинном чате AI начинает повторять себя или упорно держится за ошибку, которую ты уже исправил.

Исправление: Начни новый чат. Скопируй финальную версию задачи (не историю переписки) и задай вопрос заново. Чистый контекст = чистый ответ.

Для больших задач используй модели с большим контекстным окном. У Claude и ChatGPT есть возможность выбрать размер контекста в 1M токенов. Этого хватит для 99% задач.

Но даже 1M токенов – не бесконечность. Подробнее о context drift – в главе 7.

7. Задача за пределами возможностей AI

Симптомы: AI уверенно врёт. Выдумывает цифры, ссылки, факты.

Исправление: Признай ограничение. AI плохо работает с:

- > Точными цифрами (статистика, финансы) – проверяй
- > Свежими событиями – не знает
- > Узкоспециализированными доменами – поверхностен
- > Математикой (иногда) – считай сам

Используй Knowledge Boundary паттерн, чтобы AI честно признавал когда не знает.

> Быстрая диагностика

AI ответил плохо

- |
- | Generic/не в тему? → Добавь контекст (п.1)
- | Поверхностно? → Разбей на задачи (п.2)
- | Не тот формат? → Дай пример (п.3)
- | Тупит? → Смени модель (п.4)
- | Игнорирует часть? → Сократи промпт (п.5)
- | Повторяется? → Новый чат (п.6)
- | Выдумывает? → Ограничение AI (п.7)

> Антисовет про email

Кстати, про email'ы. Сегодня люди часто используют AI чтобы написать длинный, идеально сформулированный email. А получатель что делает?

Правильно – прогоняет его через *свой* AI, чтобы получить summary.

Два AI общаются друг с другом через тебя как прокси. Дивный новый мир. Office slop.

Если хочешь написать человеку – *пиши коротко*. 3-4 предложения. Своими словами. AI тебе для этого не нужен.

→ *Действие:* Вспомни последний раз, когда AI тебя разочаровал. Пройди по чеклисту – какой пункт был причиной?

Один промпт, который чинит 80% проблем

Когда AI ответил мимо — не переписывай свой промпт. Вставь это:

Стоп. Твой последний ответ не попал в цель.
Прежде чем отвечать заново — задай мне 3 уточняющих вопроса.

Лучший debug — заставить AI debug'ить самого себя. Он спросит именно то, чего не хватало в контексте. Ты ответишь. Следующий ответ будет в разы лучше.

Когда debug не поможет (и когда поможет, но ты не знал)

Распространённый миф: «AI не знает свежих данных.»

Это было правдой в 2024. В 2026 — нет. Большинство AI-инструментов умеют искать в интернете. Но они не делают этого по умолчанию — *нужно попросить явно*.

Найди данные за последние 2 недели по теме [X].
Используй поиск в интернете. Приложи ссылки на источники.

Я лично использую это постоянно — для ресёрча трендов, анализа конкурентов, проверки фактов. AI отлично справляется, когда знает, что ему *можно* выйти за пределы своих знаний.

А вот что реально не починишь:

- > Точные цифры без источников — AI может нагенерировать убедительную статистику из воздуха. Всегда проси ссылки. Всегда проверяй.
- > Юридические советы — иди к юристу.
- > Медицинские вопросы — иди к врачу.
- > Решения, за которые ты отвечаешь — AI помогает думать, но решаешь ты.

AI — мощный инструмент, но не оракул. Знать его границы — навык. Знать, как эти границы раздвинуть — ещё более ценный навык.

Глава 6. Практика — прокачайся за один вечер

Теория без практики — ноль. Вот два упражнения, которые превратят знания в навык.

Контекстный файл ты уже создал в главе 3 (создал же?). Теперь — паттерны и debug.

> Упражнение 1: Паттерн-рефакторинг

Время: 15 минут *Результат:* Понимание, как паттерны меняют качество ответа

Шаги:

1. Возьми задачу, которую делаешь регулярно. Примеры:
 - > Проанализировать данные
 - > Придумать идеи для контента
 - > Проверить свой план
2. Напиши «наивный» промпт — как обычно формулируют начинашки: > «Придумай идеи для LinkedIn постов.»
3. Перепиши, применив подходящий паттерн из главы 4: > «Ты — контент-стратег для B2B SaaS. > Моя аудитория: CTO и VP Engineering, 30-45 лет, скептики. > Мой стиль: прямой, с примерами из личного опыта, без эмодзи. > Задача: 5 идей для LinkedIn постов на эту неделю. > Формат: Заголовок + hook (первая строка) + краткое описание.»
4. Отправь оба промпта одному AI. Но в разных чатах. Сравни.

Как понять, что получилось: Разница между ответами очевидна. Второй — конкретнее, релевантнее, ближе к тому, что ты действительно опубликуешь.

> Упражнение 2: Prompt Debugging в бою

Время: 20 минут Результат: Навык быстрой диагностики и исправления

Шаги:

- 1. Открой свою историю чатов с AI
- 2. Найди 3 диалога, где ты был недоволен результатом
- 3. Для каждого:
 - > Определи причину по чеклисту из главы 5
 - > Перепиши промпт с исправлением
 - > Отправь исправленный промпт и оцени разницу
- 4. Заполни таблицу:

Исходный промпт	Причина провала	Исправление	Стало лучше?
...	Нет контекста (1)	Добавил роль + ситуацию	Да/Нет
...	Много задач (2)	Разбил на 2 промпта	Да/Нет
...	Нет примеров (3)	Добавил 2 примера	Да/Нет

Как понять, что получилось: Ты начинаешь видеть паттерны в своих ошибках. У большинства людей 80% проблем — одна и та же причина (обычно 1 или 2).

Впереди ещё две бонусные главы. Они про то, что происходит *после* того, как ты всё настроил. Про context drift и Projects.

Бонус 1. Context Drift — почему AI тупеет в длинных чатах

Ты настроил контекстный файл. Выучил паттерны. AI отвечает отлично. Ты доволен.

Проходит 40 минут разговора. И AI начинает... тупить. Повторяет себя. Забывает, о чём вы договорились. Упорно делает то, что ты уже запретил.

Добро пожаловать в *context drift*.

> Что происходит

У каждого AI есть *контекстное окно* — количество текста, которое он может «держать в голове» одновременно. Это как оперативная память. Когда окно заполняется, AI начинает «забывать» начало разговора.

Но есть нюанс, о котором мало кто говорит.

Эвристика: качество падает раньше, чем ты думаешь. По моему опыту и по наблюдениям сообщества, AI начинает деградировать задолго до того, как контекстное окно заполнится полностью. Примерно на середине — иногда раньше.

Это значит: даже если модель поддерживает 1 миллион токенов, это не значит, что она одинаково хорошо работает на всём протяжении. Доверяй своим ощущениям: если AI начал тупить — скорее всего, контекст перегружен.

> Context Rot

Context rot — это постепенное ухудшение качества AI по мере того, как разговор растёт:

1. *Потеря инструкций.* AI «забывает» правила из начала чата. Ты сказал «пиши коротко» — через 20 сообщений он снова пишет эссе.

2. *Дрейф стиля*. AI начинает «усреднять» стиль. Был прямой и конкретный — становится вежливый и водянистый.
3. *Петли повторений*. AI упорно возвращается к одним и тем же решениям, даже если ты их уже отверг.
4. *Галлюцинации*. Чем длиннее контекст, тем больше вероятность, что AI начнёт выдумывать.

> **Compaction — встроенная защита**

Некоторые AI-инструменты (например, Claude Code) используют *compaction* — автоматическое сжатие контекста. Когда контекстное окно заполняется, система «выжимает» из разговора ключевые факты и продолжает с ними, отбрасывая детали.

Это как конспект вместо полной записи лекции. Работает, но с потерями.

Проблема compaction: AI сам решает, что важно, а что нет. Иногда он выбрасывает именно то, что тебе нужно.

> **4 правила работы с длинными чатами**

1. Новый чат = чистый контекст

Если задача закончена — начни новый чат. Не продолжай в старом «чтобы AI помнил». Контекстный файл и так «помнит» — а детали предыдущей задачи только мешают.

2. Контекст переполнен? Саммари + новый чат

Если чувствуешь, что AI начинает тупить — скорее всего, контекст переполнен.

> В этот момент попроси AI сделать подробное саммари:

Сделай подробное саммари нашего разговора:

1. Что мы решили (ключевые решения)
2. Что осталось нерешённым (открытые вопросы)
3. Контекст, который нужно сохранить

> Проверь что в саммари вошло всё что важно

> Начни новый чат и скопируй в начало саммари из предыдущего чата

> Продолжай работать в новом чате пока не начнешь замечать дрейф контекста

3. Одна задача — один чат

Не мешай разные задачи в одном чате. «Помоги подготовиться к встрече с клиентом» и «проанализируй конкурентов» — это два разных чата. Даже если тебе лень открывать новое окно.

4. Компактные инструкции

Если используешь инструменты с compaction (Claude Code, Cursor) — помечай критичные инструкции так, чтобы они пережили сжатие. Держи их в начале контекстного файла, а не в середине разговора.

> Когда начинать новый чат

Ситуация	Действие
Задача закончена	Новый чат
AI начал повторяться	Саммари + Новый чат
AI забывает инструкции	Саммари + Новый чат
Сменил тему	Новый чат
Чат длится >30 минут	Проверь качество, возможно новый чат
Всё работает хорошо	Продолжай

→ Действие: В следующий раз, когда AI «затупит» в длинном чате — не пытайся исправить его промптом. Начни новый чат с контекстным файлом. Сравни разницу.

Что происходит прямо сейчас

AI начинает запоминать. И Claude, и ChatGPT уже умеют сохранять факты о тебе между сессиями. Ты говоришь «я работаю в финтехе» — и через неделю, в новом чате, AI это помнит.

Это не замена контекстному файлу — память пока неуправляема и непредсказуема. Но направление понятно: скоро context drift перестанет быть проблемой, потому что AI будет помнить не только текущий чат, а всю историю работы с тобой.

Пока это не произошло — используй правила из этой главы. Но следи за обновлениями: память AI развивается быстро.

Бонус 2. Projects — AI, который не забывает

Ты создал контекстный файл. Это уровень 2-3. Но что делать, когда у тебя *несколько задач*?

Один контекстный файл — для блога. Другой — для работы. Третий — для side-проекта. Каждый раз копировать нужный? Менять Custom Instructions?

Есть решение лучше: *Projects*.

> Что такое Projects

И Claude, и ChatGPT поддерживают Проекты — изолированные рабочие пространства внутри одного аккаунта. Каждый проект — это:

- > Свой контекстный файл (Project Instructions / System Prompt)
- > Свои загруженные файлы (Knowledge Base)
- > Своя история чатов
- > Свой «характер» AI

По сути, у тебя *несколько разных AI-ассистентов*, каждый настроенный под свою задачу.

> Зачем это нужно

Без Projects: Один AI знает всё обо всём. Или, точнее, ничего ни о чём — потому что контекст размывается.

С Projects: У тебя отдельный AI для каждой роли:

- > «Блог-ассистент» — знает твой стиль, аудиторию, tone of voice
- > «Рабочий аналитик» — знает твой продукт, метрики, конкурентов
- > «Карьерный коуч» — знает твоё резюме, цели, рынок
- > «Помощник по автоматизации» — знает твои задачи, инструменты, уровень

Каждый помнит контекст между сессиями. Не надо объяснять заново.

> Как устроено

Claude Projects (claude.ai)

1. Зайди в `claude.ai` → боковая панель → «Projects»
2. Создай новый проект (например: «Мой блог»)
3. Добавь *Project Instructions* — это твой контекстный файл для этого проекта
4. Добавь файлы в *Knowledge* — примеры текстов, гайдлайны, данные
5. Все чаты внутри проекта автоматически видят эти инструкции и файлы

Что положить в *Knowledge*:

- > Примеры твоих лучших текстов (для блога)
- > Документацию продукта (для работы)
- > Своё резюме и *cover letter* (для карьеры)
- > Учебные материалы (для обучения)

ChatGPT Projects

1. Зайди в ChatGPT → боковая панель → «Projects»
2. Создай проект
3. Добавь *Instructions* и *Files* — работает аналогично Claude
4. Все чаты в проекте используют эти инструкции

> 5 проектов, с которых стоит начать

Проект	Instructions (суть)	Knowledge (файлы)
Мой блог	Tone of voice, аудитория, стиль, антипаттерны	5-10 лучших постов
Работа	Роль, продукт, метрики, команда	Доки продукта, OKR
Карьера	Текущая роль, цели, навыки	Резюме, LinkedIn
Обучение	Тема, уровень, стиль обучения	Учебные материалы
Side-проект	Описание, стек, стадия	Код, wireframes, заметки

> Антисовет

Не создавай 20 проектов. Начни с 2-3 самых частых задач. Добавляй по мере необходимости. Лучше 3 хорошо настроенных проекта, чем 15 пустых.

> Projects vs контекстный файл

Контекстный файл	Projects
Что это	Текст с инструкциями
Рабочее пространство	Где живёт
Один на всё	По одному на задачу
Что помнит	Кто ты и как работать
1. загруженные файлы + историю	Когда использовать
Одна основная задача	Несколько разных задач

Projects – это контекстный файл на стероидах. Системный промпт говорит «кто ты». Projects говорит «кто ты в этой роли, с этими данными, в этом проекте».

→ *Действие:* Создай один проект в Claude или ChatGPT. Начни с самой частой задачи. Добавь инструкции из своего контекстного файла + 3-5 релевантных файлов. Поработай неделю — оцени разницу.

Важный нюанс: файлы в проекте ≠ файлы в голове

Ты загрузил в проект 15 файлов. Кажется, что AI теперь «знает» их все. Это не совсем так.

AI не перечитывает все файлы при каждом запросе. Он ищет релевантные фрагменты — и иногда промахивается. Чем больше файлов, тем выше шанс, что нужный затеряется.

Правило хорошего тона: В начале нового чата подскажи, на какие файлы обратить внимание.

В этом чате работаем с файлами:
- `tone-of-voice.md` — мой стиль для текстов
- `audience.md` — описание аудитории
Остальные файлы проекта пока не нужны.

Это как сказать коллеге: «Открой вот эти два документа, остальное сейчас не важно.» Простое действие — но качество ответов вырастает заметно.

Что дальше

Ты прочитал гайд и сделал упражнения. Теперь ты:

- > Понимаешь разницу между промптом и контекстом
- > Знаешь свой уровень и как его поднять
- > Имеешь контекстный файл, который делает AI персональным
- > Владеешь 10 паттернами для разных задач
- > Умеешь диагностировать и чинить плохие ответы

Следующий шаг: Используй. Каждый день. Через неделю ты не вспомнишь, как работал без контекста.

Это Playbook #1 — фундамент. Следующие плейбуки выходят на postnikov.ai.

Если хочешь систему, а не разовый совет → hellonew.pro