ПРАКТИКА ИИ · 02

РАБОТАТЬ С ИИ

От постановки задачи до рабочего помощника

Марина Лямкина

Марина Лямкина. «Работать с ИИ».
Серия «Практика ИИ». Книга 02.
Электронное издание.
© Марина Лямкина, 2026.
Все права защищены. Не допускается воспроизведение и распространение текста и иллюстраций целиком или частично без разрешения правообладателя, кроме случаев, предусмотренных законом.

О СЕРИИ «ПРАКТИКА ИИ»

«Практика ИИ» — это серия коротких прикладных книг о том, как обычному человеку и специалисту без технического бэкграунда разобраться в искусственном интеллекте и начать использовать его в работе и жизни без лишнего шума, мифов и усложнений.

Первая книга серии — **«Понять ИИ»** — была о базовых вещах: что такое модель, контекст, память, поиск, инструменты, почему ИИ может ошибаться и как с ним вообще правильно разговаривать.

Вторая книга — **«Работать с ИИ»** — уже не про устройство, а про практику.

Она о том, как перейти от случайных запросов к нормальной рабочей системе: правильно ставить задачу, работать итерациями, сохранять удачные решения, собирать свои заготовки и постепенно превращать ИИ из игрушки в полезного помощника.

Эта серия устроена просто:

сначала — понять,
потом — научиться работать,
дальше — использовать осмысленно и под свою реальность.

Если после первой книги читатель перестаёт бояться непонятных слов,

то после второй — уже начинает работать увереннее и с куда большей отдачей.

ОТ АВТОРА

Почему эта книга начинается не с промпта

Есть момент, который я постоянно наблюдаю, когда люди начинают пользоваться ИИ всерьёз.

Человек ставит задачу, получает ответ, читает его и почти сразу выносит вердикт:

— Нет, вообще не то.

Иногда действительно не то. Иногда даже настолько не то, что хочется закрыть чат и больше к этой теме сегодня не возвращаться.

Но мне в этот момент всегда интереснее другое: что человек сделает дальше?

Попросит другую нейросеть выполнить ту же задачу? Начнёт переписывать запрос с нуля? Пойдёт искать в интернете «правильный промпт»? Решит, что ИИ подходит только для простых текстов? Или останется в этом же разговоре и начнёт разбираться, что конкретно ему не понравилось?

Мне кажется, именно здесь проходит одна из самых важных границ в работе с ИИ.

Пока мы относимся к первому ответу как к экзамену для нейросети, всё довольно случайно. Получилось — инструмент хороший. Не получилось — плохой. Сегодня одна модель «умнее», завтра другая. Один и тот же человек может за неделю несколько раз полностью изменить мнение об ИИ просто потому, что в одной задаче результат случайно оказался очень удачным, а в другой — совершенно бесполезным.

Но в какой-то момент появляется другое отношение.

Первый ответ перестаёт быть итогом.

Он становится началом работы.

И вот здесь всё меняется.

Я могу посмотреть на текст и сказать: содержание правильное, но я так не разговариваю. Или: начало хорошее, дальше пошли банальности. Или: ничего не переписывай, потерялась только одна важная мысль. Или вообще: я пока не понимаю, что именно мне не нравится, давай сначала разберёмся.

После этого мы делаем ещё одну версию.

Иногда вторую.

Иногда пятую.

Иногда на третьей понимаем, что первая была лучше, и возвращаемся назад.

Иногда в процессе выясняется, что проблема вообще была не в ответе. Я сама до первого варианта не могла нормально сформулировать, чего хотела.

Это очень важное свойство работы с ИИ, которое почему-то легко недооценить.

Мы много говорим о качестве запроса. Но человек далеко не всегда способен заранее описать идеальный результат. Особенно если задача новая, творческая, сложная или просто плохо определена.

Попробуйте до первой версии дизайна точно описать, насколько «современно» должен выглядеть слайд. До первого черновика книги — где именно текст должен стать менее официальным. До анализа данных — какие связи окажутся действительно интересными. До новой структуры документа — какой раздел вдруг станет очевидно лишним.

Иногда критерий появляется только тогда, когда есть с чем сравнивать.

Именно поэтому эта книга не будет начинаться с формулы идеального промпта.

В первой книге серии мы уже разобрались, что такое промпт, контекст, модель, поиск, файлы и инструменты. Там же я специально остановилась

в тот момент, когда вопрос «что это такое?» начал превращаться в другой: «и что теперь с этим делать?»

Теперь будем отвечать именно на него.

Причём мне хочется говорить не о красивых демонстрациях возможностей ИИ, а о довольно обычной работе. О той, где есть исходные материалы, дедлайны, старые версии документов, таблицы с пропусками, разговоры, в которых половину нужного так и не спросили, тексты, которые уже почти хорошие, и решения, за которые потом отвечает не нейросеть, а человек.

В такой работе редко существует один промпт, после которого можно больше ничего не делать.

Гораздо чаще есть процесс.

Вы ставите задачу. Получаете первую версию. Смотрите, где ИИ попал, а где нет. Уточняете. Исправляете. Сохраняете удачное. Возвращаете потерянное. Иногда добавляете новые материалы. Иногда понимаете, что данных вообще не хватает. Проверяете результат. И только потом решаете, что работу можно закончить.

А дальше происходит ещё одна вещь, которую я считаю особенно полезной.

Если задача повторяется, весь этот путь можно не проходить заново.

После того как мы несколько раз исправили результат и наконец получили именно то, что нужно, можно посмотреть назад и собрать из проделанной работы новую инструкцию.

Не ту, с которой начали.

А ту, которую теперь уже умеем сформулировать.

В ней появятся требования, которых в начале вообще не было. Ограничения, про которые мы вспомнили после второй версии. Критерии, которые смогли назвать только тогда, когда увидели неудачный результат. Формат, который оказался удобнее первоначального. И вещи, которые ИИ несколько раз пытался менять, хотя их нужно было сохранять.

Так из работы постепенно рождается хороший промпт.

Потом его можно проверить в новом чате, на новом материале. Если результат воспроизводится — прекрасно. Значит, мы нашли способ решать эту задачу не случайно.

Если задача начинает повторяться постоянно, можно пойти ещё дальше и сделать помощника, которому часть этих правил уже не придётся объяснять каждый раз.

Мне нравится именно такая последовательность.

Не начинать с вопроса:

— Какого агента нам сделать?

А сначала хотя бы несколько раз нормально сделать саму работу.

Не пытаться сразу написать универсальную инструкцию для процесса, который мы ещё ни разу не прошли до конца.

Не автоматизировать собственную неопределённость.

Сначала разобраться, как получается хороший результат.

Потом закрепить найденный способ.

Эта книга будет именно об этом пути.

От первой попытки, после которой хочется сказать «вообще не то», — до работы, которую уже можно повторить.

И начнём с самого простого момента.

Ответ уже перед нами.

Он нам не нравится.

Что теперь?

ГЛАВА 1

ЧТО ИМЕННО ЗДЕСЬ НЕ ТАК?

Итак, ответ уже перед нами. Он не устраивает. Иногда понятно почему: перепутана цифра, пропущен важный пункт, неверно понята задача. С этим относительно просто. Гораздо интереснее другой случай — когда всё вроде бы нормально, а использовать результат не хочется.

Текст грамотный, но какой-то чужой. Презентация аккуратная, но смотреть на неё тяжело. Анализ выглядит убедительно, однако после десяти страниц всё ещё непонятно, что именно мы узнали. Сценарий разговора содержит все нужные вопросы, но сотрудник теряется в нём во время звонка. Коммерческое предложение формально собрано правильно, а клиенту его отправлять рано.

В такой момент очень легко написать:

Плохо. Переделай.

ИИ переделает. И вот здесь начинается проблема: он знает, что результат нам не понравился, но не знает почему. Поэтому следующая версия становится не столько исправлением первой, сколько новой попыткой угадать наше ожидание.

Если повезёт, станет лучше. Если нет — изменится то, что было хорошим, а настоящая проблема останется.

Я сама много раз попадала именно в эту ловушку. Особенно когда вижу результат и первая реакция эмоциональная: «Нет, вообще не то». Иногда эта реакция абсолютно точная. Но как рабочая инструкция она почти бесполезна.

Поэтому сейчас, если задача для меня важна, я стараюсь не начинать с переделки. Сначала мне нужно понять, что именно я хочу исправить.

> ПРАВИЛО. **Не переделывайте результат, пока не поняли, что именно в нём не устраивает. Чем точнее определена проблема, тем меньше хорошего будет случайно разрушено следующей версией.**

При этом совсем не обязательно уже уметь профессионально назвать проблему. Можно честно сказать ИИ:

Мне не нравится этот текст, но я пока не могу точно объяснить почему. Ничего не переписывай. Сначала помоги мне разобрать результат и предложи возможные причины.

Это нормальная задача. Более того, иногда она оказывается гораздо полезнее просьбы сразу всё исправить.

Представим текст, про который я могу сказать только одно:

Он слишком ИИшный.

Для меня эта формулировка что-то означает. Для другого человека — другое. Для модели — ещё десятки возможных вариантов.

Но если начать смотреть на конкретный текст, ощущение довольно быстро раскладывается на наблюдаемые вещи.

Например, выясняется, что почти каждый абзац заканчивается маленьким выводом. В тексте слишком много одинаковых конструкций: «не это, а то», «важно не только..., но и...». Короткие предложения используются настолько часто, что вместо естественного ритма появляется постоянное искусственное ударение. Автор всё время что-то подытоживает, объясняет читателю очевидное и как будто боится оставить мысль без финальной таблички «что нужно запомнить».

Или проблема вообще другая. Текст слишком гладкий. В нём нет слов и оборотов, которыми конкретный автор обычно пользуется. Исчезли сомнения, раздражение, наблюдения, неровности живой речи. Получился профессиональный текст, который мог бы принадлежать кому угодно.

Теперь «ИИшно» перестало быть просто ощущением. Появились критерии, с которыми можно работать.

В таком случае я могу написать:

Теперь я понимаю, что именно не нравится. Основные проблемы: слишком много коротких акцентных предложений, постоянные противопоставления и выводы после каждой мысли. Кроме того, пропал

мой обычный разговорный тон. Содержание пока не меняй. Перепиши только с точки зрения этих замечаний.

Получаем новую версию и снова её оцениваем. Если проблема исправлена не полностью, уточняем ещё раз. Если вместе с исправлением появилось что-то новое и неудачное, разбираемся уже с этим.

И вот это последовательное движение от одной версии к следующей дальше я буду называть **итерационной работой**.

Итерация — один цикл работы с результатом: мы получаем версию, смотрим, что в ней получилось и что нет, вносим уточнение и получаем следующую.

Само слово звучит несколько технически, а на практике описывает совершенно обычную вещь. Мы не пытаемся каждый раз заново решить всю задачу. Мы постепенно доводим уже полученный результат.

При этом новая генерация сама по себе ещё не означает новую полезную итерацию. Если после каждого ответа писать только «не то, попробуй ещё раз», мы скорее перебираем варианты в надежде, что один случайно понравится. В нормальной итерационной работе после каждой версии появляется новая информация: мы лучше понимаем, что нужно сохранить, что изменить, какой критерий оказался важен или чего не учли в начале.

Поэтому несколько итераций сами по себе совсем не говорят о том, что первый промпт был плохим или ИИ «не справился». Для сложной работы это часто нормальный способ получить нужный результат.

> ПРАВИЛО. **Хорошая итерация уменьшает неопределённость. После каждой следующей версии мы должны чуть лучше понимать, что именно считаем хорошим результатом и как к нему прийти.**

Та же логика работает с любым расплывчатым требованием.

«Сделай профессиональнее».

Что именно должно измениться? Мне нужен более официальный язык? Более точные формулировки? Меньше рекламных обещаний? Больше цифр? Чётче логика?

«Сделай современнее».

Современнее по сравнению с чем? Нужно убрать визуальный шум? Сократить текст? Изменить язык? Отказаться от устаревшей структуры?

«Сделай глубже».

Нужны дополнительные данные? Другие объяснения? Проверка альтернативных гипотез? Или ответ просто получился слишком общим?

До появления первой версии человек часто не знает этих деталей. Именно поэтому результат полезен даже тогда, когда он не подошёл: на нём можно увидеть собственные требования.

И отсюда появляется следующее правило.

> ПРАВИЛО. **Общая оценка должна постепенно превращаться в наблюдаемые критерии. «Плохо», «слабо», «непрофессионально», «слишком ИИшно» — нормальная первая реакция, но для следующего шага нужно понять, какие конкретные признаки создают это ощущение.**

Иногда ИИ может помочь эти признаки найти.

Мне нравится в таких случаях специально останавливать его:

Пока ничего не меняй. Разбери, почему этой презентацией неудобно пользоваться на совещании. Оцени не красоту слайдов, а ситуацию: руководитель видит каждый слайд около минуты и должен быстро понять вывод.

Здесь сразу появляется важная привязка. Не «удобная презентация» вообще, а удобная для конкретного действия.

Предположим, система отвечает, что на большинстве слайдов слишком много равнозначных элементов, основной вывод не выделен, таблицы требуют долгого чтения, а часть текста фактически повторяет то, что будет говорить выступающий.

Теперь можно решать, с какими замечаниями согласны, и только после этого переделывать.

То же самое со сценарием разговора. Если написать «сделай сценарий удобнее», ИИ вполне может решить, что удобнее означает короче. Он сократит вопросы, объединит блоки, уберёт часть подсказок. Документ станет компактнее, но сотруднику во время звонка всё равно будет трудно им пользоваться.

Потому что проблема была не в количестве текста как таковом. Менеджер должен одновременно слушать клиента, говорить и быстро находить глазами следующий вопрос. Значит, настоящий критерий выглядит примерно так:

Содержание вопросов нас устраивает. Не сокращай его. Проблема в том, что во время разговора сложно быстро найти основной вопрос. Перестрой только навигацию: основной вопрос должен визуально находиться сразу, а пояснения и возможные реакции клиента должны быть вторым уровнем.

Теперь модель понимает не только, что нужно изменить, но и зачем.

Этот принцип я бы вообще переносила на большинство рабочих задач. Результат оценивается не сам по себе. Он существует для какого-то следующего действия.

Письмо должно привести к понятному ответу клиента. Справка — помочь принять решение. Сценарий — использоваться непосредственно в разговоре. Аналитика — ответить на конкретный управленческий вопрос. Инструкция — позволить выполнить действие без автора инструкции рядом.

Когда мы знаем, что человек будет делать с результатом, слова «хорошо» и «плохо» становятся намного содержательнее.

> ПРАВИЛО. **Качество результата лучше оценивать через его использование. Не спрашивайте только: «Хорошо ли получилось?» Спрашивайте: «Сможет ли человек сделать с этим то, ради чего материал вообще создавался?»**

Но даже после того, как проблема найдена, остаётся довольно неприятная ловушка.

Мы объясняем ИИ, что нужно исправить, и забываем сказать, что уже получилось хорошо.

Представим, что первая версия коммерческого предложения вполне нормальная. Структура подходит, данные клиента собраны правильно, предложение сформулировано понятно, техническая часть не вызывает вопросов. Не нравится только вступление — оно длинное и слишком официальное.

Если написать:

Сократи вступление и сделай КП живее,

новая версия может неожиданно изменить весь документ. Заголовки станут другими, поменяется порядок аргументов, появятся более энергичные рекламные фразы, заключение тоже «оживёт», хотя его никто не просил трогать.

Модель не обязательно делает что-то нелогичное. Просто мы не обозначили границу работы.

Поэтому обратная связь должна содержать не только критику:

Структура документа, аргументы, технический раздел и заключение подходят — их не меняй. Работаем только со вступлением. Оно слишком длинное и повторяет то, что дальше раскрывается подробнее. Сократи примерно вдвое, не добавляя новых тезисов.

Теперь понятно, где заканчиваются полномочия этой итерации.

Мне кажется, это один из наиболее недооценённых навыков работы с ИИ: уметь сохранять уже хорошее.

Мы очень много говорим о том, как правильно попросить модель что-то сделать, но гораздо реже — как попросить её не переделывать то, что уже сделано нормально. А именно на этом часто разваливается длинная работа.

Была удачная фраза — потерялась после сокращения. Был хороший пример — исчез после перестройки структуры. Были проверенные цифры — округлились при оформлении новой таблицы. Была осторожная формулировка — стала категоричной после просьбы «усилить вывод».

Последний случай особенно опасен в деловых материалах.

Клиент сказал:

Рассмотреть закупку сможем не раньше ноября.

После редакции появляется:

Клиент планирует закупку в ноябре.

Фраза стала короче и удобнее для отчёта. Но значение изменилось.

Или:

Мы рассчитываем завершить проект до 15 сентября.

После просьбы сделать текст увереннее:

Проект будет завершён до 15 сентября.

Вроде совсем небольшая стилистическая правка. Только степень обязательства уже другая.

Поэтому в важных документах я стараюсь заранее определить, что можно редактировать свободно, а что является содержательной константой.

Например:

Улучшай язык и структуру, но не меняй цифры, даты, договорённости, цитаты, фактические выводы и степень уверенности формулировок.

Или ещё конкретнее:

Все суммы, сроки и обязательства сохраняй по смыслу без изменений. Если для сокращения нужно убрать один из них, сначала сообщи об этом, а не удаляй самостоятельно.

Так мы постепенно приходим к ещё одному правилу.

> ПРАВИЛО. **В каждой значимой итерации нужно определить не только область изменений, но и область сохранения: что уже утверждено, что можно менять, а что нельзя трогать без отдельного согласования.**

Это правило особенно хорошо работает на больших текстах, презентациях, аналитике, сценариях и документах, которые проходят несколько кругов доработки.

И здесь возникает ещё одна причина, по которой результат иногда становится хуже после вполне разумной правки: мы пытаемся исправить слишком много разных вещей одновременно.

Допустим, есть большая статья. Мы просим:

Сократи её, убери повторы, сделай язык живее, усили аргументацию, проверь факты, исправь логику, сохрани авторский стиль и вычитай ошибки.

Такой запрос выглядит очень обстоятельно. Но если после выполнения статья стала хуже, разобраться в причине трудно.

Что именно произошло? При сокращении исчезло важное объяснение? Во время работы над стилем изменилась степень категоричности? «Усиление аргументации» породило новый тезис? Фактчекинг привёл к переписыванию абзаца, который вообще не нужно было трогать?

Чем сложнее материал, тем полезнее иногда не пытаться получить финальную версию одним движением.

Например, сначала посмотреть только на содержание: есть ли повторы, противоречия, логические дыры и лишние фрагменты. После принятия содержательных решений отдельно поработать с языком. И только когда текст уже не планируется существенно перестраивать, делать финальную корректуру.

Я не считаю это универсальным алгоритмом, которому нужно следовать всегда. Маленькое письмо вполне можно исправить за один проход. Но если работа важная и состоит из нескольких разных типов решений, разделение итераций делает процесс гораздо понятнее.

То же относится к таблицам. Нет большого смысла одновременно исправлять исходные данные, считать показатели, объяснять причины и формулировать управленческие рекомендации. Если потом обнаружится ошибка в исходной таблице, придётся пересобирать всю цепочку.

Сначала нужно понять, что произошло с данными. Потом убедиться, что расчёт корректен. Только после этого обсуждать смысл результата.

То есть итерации нужны не только потому, что ИИ «не угадал с первого раза». Иногда мы сознательно строим работу поэтапно, потому что одни решения зависят от других.

> ПРАВИЛО. **Одна итерация должна решать понятную проблему. Чем больше разных типов изменений мы смешиваем одновременно, тем сложнее контролировать, почему следующая версия стала лучше или хуже.**

Есть ещё один приём, которым я часто пользуюсь, если вообще не могу нормально сформулировать критерий: сравнение.

Предположим, есть два текста. Старый написан мной — он неидеальный, но узнаваемый. Новый от ИИ — чище, логичнее, однако звучит чужим.

Можно долго пытаться объяснить модели собственный стиль. А можно дать оба текста и сказать:

Не переписывай их. Сравни и покажи, за счёт каких конкретных особенностей первый текст больше похож на живую авторскую речь: длина и устройство предложений, выбор слов, степень категоричности, количество пояснений, переходы между мыслями, отношение к читателю. Отдельно покажи, что во втором тексте делает голос более обезличенным.

Теперь мы работаем не с абстрактным понятием «мой стиль», а с различиями между двумя реальными материалами.

То же самое можно делать с презентациями, отчётами, коммерческими предложениями и даже аналитикой.

Вот старый отчёт, которым руководитель действительно пользовался. Вот новый, который выглядит гораздо современнее, но оказывается бесполезным. Что именно потерялось?

Вот сценарий разговора, с которым менеджеру удобно. Вот новый, более подробный. Почему старый работает лучше во время звонка?

Вот два результата анализа. Один помог принять решение, другой просто содержит больше текста. В чём структурная разница?

Сравнение позволяет довольно быстро извлекать правила из практики. Именно извлекать, потому что зачастую они уже существуют, просто до этого никто их не записывал.

Компания годами делает хорошие предложения клиентам, но никогда не формулировала, почему одни считаются хорошими. Руководитель сразу видит слабую презентацию, но критерии существуют только у него в голове. Редактор чувствует, что текст «не авторский», но никогда не превращал это ощущение в набор проверяемых признаков.

Работа с ИИ заставляет вытаскивать эти скрытые критерии наружу. И это, пожалуй, гораздо более интересный эффект, чем просто ускорение подготовки документов.

После нескольких итераций у нас постепенно появляются собственные правила.

Например, мы выяснили, что в протоколе встречи никогда нельзя превращать обсуждение в принятое решение. В коммерческом предложении цена берётся только из актуального прайса. В сценарии звонка основной вопрос должен быть виден сразу. В авторском тексте мы не используем постоянные искусственные противопоставления и не дробим каждую мысль на короткие предложения. В анализе гипотезы всегда отделяются от того, что действительно следует из данных.

Сначала каждое из этих требований возникает как реакция на конкретную неудачную версию. Потом таких реакций становится несколько. И в какой-то момент мы обнаруживаем, что перед нами уже не

набор случайных замечаний, а описание того, как для нас выглядит хороший результат.

Это последний принцип этой главы и один из самых важных для всей книги.

> ПРАВИЛО. **Не теряйте удачные замечания после исправления конкретной версии. Если одно и то же требование важно и для следующих задач, оно должно перейти из разовой правки в постоянное правило работы.**

Так постепенно меняется сама логика взаимодействия с ИИ. Сначала мы просто реагируем на результат: нравится или не нравится. Потом учимся находить конкретную проблему. Каждая следующая итерация помогает сделать критерии точнее. Следом мы определяем границы изменений и фиксируем то, что уже хорошо. Через несколько задач начинают повторяться одни и те же требования, и мы наконец можем их записать.

Именно из них позднее получится хороший рабочий промпт — не из универсального шаблона, найденного заранее, а из того, что мы действительно выяснили о собственной работе.

Но прежде чем собирать такую инструкцию, нужно решить ещё одну проблему. Даже при очень точной обратной связи новая версия иногда оказывается хуже предыдущей. И чем больше итераций мы проходим, тем легче потерять хороший вариант, к которому потом хочется вернуться.

Следующая глава как раз будет об этом: как вести такую работу так, чтобы новая правка не уничтожала уже найденный результат.

ГЛАВА 2

ВЕРНИ ПРЕДЫДУЩУЮ ВЕРСИЮ

Есть момент, который почти неизбежно наступает после нескольких итераций. Мы уже не пытаемся понять, что именно нам не нравится, — с этим разобрались. Дали нормальную обратную связь, получили следующую версию и увидели, что проблема действительно исправлена.

Только вместе с ней исчезло что-то хорошее.

Например, мы работаем над выступлением. В первой версии слишком много материала, зато получилось отличное начало. Просим сократить. Вторая версия становится заметно лучше по объёму, но начало ИИ тоже переписывает. Просим вернуть прежнее начало. Возвращает, но одновременно меняет пример в середине. Пример теперь хуже. Его снова правим, после чего обнаруживаем, что из заключения исчезла фраза, которую вообще никто не просил трогать.

И в какой-то момент появляется:

Верни предыдущую версию.

Фраза кажется совершенно понятной. Пока предыдущая версия действительно одна.

Через двадцать сообщений всё становится интереснее.

Что именно мы называем предыдущей? Ту, которая была непосредственно перед последней правкой? Ту, где ещё оставалось старое вступление? Ту, которую вчера назвали хорошей? Или вариант, который был три итерации назад и который мы сейчас внезапно вспомнили?

Человек ещё может примерно держать эту историю в голове. ИИ тоже видит предыдущую переписку, пока она доступна в текущем контексте. Но я бы не строила серьёзную работу на надежде, что и мы, и система через час одинаково понимаем выражение «тот вариант, который был раньше».

Поэтому примерно в этот момент появляется новый рабочий навык: кроме самого результата, приходится управлять его версиями.

Под **версией** здесь я буду понимать просто сохранённый вариант работы на определённом этапе. Никакой сложной системы контроля версий нам пока не нужно. Версия может называться «первый черновик», «после сокращения», «согласованное содержание», «финал перед отправкой». Смысл не в названии, а в том, чтобы в нужный момент можно было точно понять, к какому состоянию работы мы хотим вернуться.

И есть одна вещь, которую лучше принять сразу: новая версия не обязана быть лучше предыдущей.

Само слово «доработка» создаёт ощущение движения вперёд. Мы исправили недостаток, значит, стало лучше. На практике каждая итерация может улучшить один элемент и ухудшить другой. Особенно если результат большой: текст, презентация, аналитический отчёт, сценарий, коммерческое предложение.

Именно поэтому последняя версия — это только текущее состояние работы. Не победитель конкурса всех предыдущих вариантов.

Представим, что в первой версии презентации нам понравилась логика. Во второй мы улучшили дизайн. В третьей сократили текст, но вместе с сокращением потеряли два важных аргумента. В четвёртой аргументы вернули, однако поменялся порядок слайдов и пропала удачная логика первой версии.

Если просто продолжать переделывать четвёртую, мы можем ещё долго ходить по кругу. Иногда разумнее остановиться и признать: хороший итог уже существует, просто его части находятся в разных вариантах.

> ПРАВИЛО. **Последняя версия не считается лучшей автоматически. Новая итерация имеет смысл только в сравнении с тем, что было до неё: что стало лучше, что осталось прежним и что случайно ухудшилось.**

Это кажется элементарным, но на практике люди часто оценивают только то, что сейчас находится перед глазами. Старый текст уже выше в переписке, поэтому он психологически как будто перестал существовать.

А потом возникает фраза:

— Там раньше было лучше написано.

Где именно «там» — уже приходится искать.

И вот здесь я хочу буквально объяснить, что имею в виду, когда дальше буду писать: **«сохраните хорошую версию»**.

Не просто скажите ИИ:

Считаем этот вариант базовым.

Это полезно для текущего разговора, но для важной работы этого мало.

Если перед вами получился результат, который вы действительно не хотите потерять, сохраните его **вне чата**. Скопируйте текст в Word и сохраните отдельным файлом. Если сервис позволяет скачать готовый документ — скачайте его. Таблицу, презентацию, PDF или другой готовый файл тоже сохраните отдельно. Можно положить его в обычную папку на компьютере или в облачное хранилище, например на Яндекс.Диск.

То есть действие совершенно буквальное:

> **получили хорошую версию →**
>
> **сохранили отдельным файлом →**
>
> **только потом продолжаем экспериментировать.**

Не рассчитывайте на то, что «это же всё осталось в переписке».

Да, осталось. Но переписка — плохой архив для важного документа. Через несколько итераций придётся искать нужный вариант среди похожих текстов. Через несколько дней можно вообще забыть, в каком именно чате он был. А если вы сильно перерабатывали материал, восстановить нужную версию по памяти разговора оказывается гораздо сложнее, чем просто открыть сохранённый файл.

Например, мы подготовили коммерческое предложение. В целом оно уже устраивает, но хотим ещё попробовать другой стиль вступления. Я бы сначала сохранила файл:

КП_клиент_01_рабочая.docx

После согласования содержания:

КП_клиент_02_содержание_согласовано.docx

После финальной проверки:

КП_клиент_03_к_отправке.docx

Не нужно копировать именно такую систему названий. Можно назвать файлы совершенно иначе. Главное, чтобы через неделю человек сам без расследования понял, какой из них открывать.

С книгой то же самое. Если глава наконец получилась хорошо, а дальше я собираюсь существенно менять её структуру, сначала сохраню хорошую версию отдельно. Тогда эксперимент уже не страшен. Даже если новая редакция окажется хуже, исходный вариант никуда не делся.

> ПРАВИЛО. **Всё, что вы действительно боитесь потерять, должно существовать вне текущего чата. Хорошую версию нужно не только назвать базовой, но и физически сохранить: в Word, отдельном файле или облачном хранилище, например на Яндекс.Диске.**

Здесь важно не смешивать два разных действия.

Первое — сохранить материал для себя.

Второе — сообщить ИИ, что именно эта версия становится точкой отсчёта для дальнейшей работы.

После сохранения я могу написать:

Этот вариант я сохранила отдельно. Считаем его базовой версией. Дальше можем пробовать изменения, но если результат станет хуже, возвращаемся именно к этой базе.

Теперь у меня есть реальный файл, а в диалоге есть договорённость.

Одно не заменяет другое.

Особенно это полезно перед правками, которые способны сильно изменить результат. Например:

Полностью перестрой структуру.

Сократи в два раза.

Попробуй совершенно другой стиль.

Собери новый вариант из этих материалов.

Перед такими экспериментами базовую версию лучше сохранить почти автоматически.

Мне вообще нравится простое правило: **чем смелее следующая правка, тем важнее сначала оставить себе безопасную точку возврата**.

В какой-то момент версий начинает становиться несколько, и возникает следующая проблема: разные их части имеют разный статус.

Предположим, мы готовим отчёт. Цифры уже проверили. Выводы пока обсуждаем. Первый раздел текста утверждён. Заключение ещё черновое.

Если просто отправить ИИ весь документ и попросить:

Улучши отчёт,

для модели это один большой материал, который можно переработать.

Для нас — нет.

Внутри уже есть вещи, которые трогать нельзя.

Поэтому на больших задачах я стараюсь явно обозначать статус отдельных элементов:

Расчёты уже проверены. Их не пересчитывай и не меняй.

Первый раздел утверждён. Его сохраняем без смысловых изменений.

Второй раздел пока черновой, там можно предлагать другую структуру.

Заключение пока вообще не утверждено — с ним можно работать свободнее.

В результате документ перестаёт быть одним сплошным текстом, который каждый раз заново генерируется целиком. У него появляется внутренняя структура решений: это уже готово, это можно менять, здесь ещё думаем.

Особенно важно делать это с цифрами, фактами и договорённостями.

Представим, что мы уже проверили расчёт экономического эффекта. Сверили исходные данные, нашли ошибку, исправили, получили итоговую цифру. После этого просим:

Теперь подготовь для руководителя краткое объяснение расчёта.

Я не хочу, чтобы при подготовке объяснения ИИ ещё раз пересчитывал экономику каким-то своим способом. Работа с числами закончена. Теперь начинается работа с объяснением.

Поэтому напишу прямо:

Итоговая цифра уже проверена и утверждена. Не пересчитывай её. Используй как исходное условие следующего этапа.

То же самое с документами.

Мы проверили дату договора. Установили правильное название продукта. Согласовали срок. Эти элементы уже не должны свободно меняться только потому, что следующий запрос посвящён стилю.

Мне очень нравится этот момент в работе: проверенный элемент как будто меняет свой статус. Пока мы выясняли цифру, она была частью задачи. После проверки она становится **исходным условием** для дальнейшей работы.

> ПРАВИЛО. **То, что уже проверено или утверждено, нужно перестать генерировать заново. Проверенные цифры, факты, решения и критичные формулировки становятся исходными условиями следующих шагов.**

Это заметно снижает количество странных ситуаций, когда в пятой версии документа внезапно появляется другая сумма или уже согласованная формулировка неожиданно становится более категоричной.

Есть ещё одна полезная привычка: сохранять версию именно в момент перехода от одного типа работы к другому.

Например, сначала мы работали над содержанием главы. Разобрались с логикой, убрали лишние блоки, добавили недостающий пример. В какой-то момент содержание нас устраивает.

Вот здесь я бы остановилась.

Сохранила файл.

И только потом переходила к языку.

Почему не продолжить всё в том же режиме?

Потому что стилистическая редактура тоже способна изменить содержание. Мы уже видели это в первой главе. Фраза становится легче, а смысл немного сдвигается. Убирается повтор — вместе с ним исчезает важный акцент. Модель решает улучшить переход и добавляет промежуточный вывод, которого раньше не было.

Если содержательная версия сохранена, я могу спокойно сравнить результат.

С данными тот же принцип. Сначала завершили проверку исходных данных — сохранили. Затем сделали расчёт — зафиксировали результат. Потом уже переходим к интерпретации и оформлению.

С презентацией: сначала согласовали содержание и последовательность слайдов, потом работаем над текстами и визуальной частью.

Так гораздо легче понять, на каком этапе появилась проблема.

> ПРАВИЛО. **Хорошая точка для сохранения версии — завершение отдельного этапа работы. Содержание согласовано — сохраняем. Расчёт проверен — сохраняем. Структура утверждена — сохраняем. После этого можно безопаснее переходить к следующему типу изменений.**

Это не означает, что теперь любой документ должен существовать в двадцати семи файлах.

У небольшой задачи вполне может быть всего два сохранённых состояния: рабочее и финальное.