

# Работа на автопилоте

Как поручить нейросети  
ежедневные задачи

*Больше  
важного*

✓ Планирование

✓ Анализ

✓ Контент

✓ Коммуникации

✓ Отчеты



AI  
WORKS  
FOR YOU

АВТОПИЛОТ



Задачи



Нейросеть



Результат



МЕНЬШЕ  
РУТИНЫ  
БОЛЬШЕ  
ВОЗМОЖНОСТЕЙ

ИДЕИ  
РАЗВИТИЕ  
СВОБОДА

ГЛАВНОЕ  
ВПЕРЕДИ

Марк Тьюрин

**Работа на автопилоте:  
Как поручить нейросети  
ежедневные задачи**

«Автор»

2026

**Тьюрин М.**

Работа на автопилоте: Как поручить нейросети ежедневные задачи  
/ М. Тьюрин — «Автор», 2026

Вы открываете почту, разбираете документы, планируете встречи и снова откладываете важное — не из-за лени, а потому что день разрывают повторяющиеся задачи. «Работа на автопилоте» показывает, как превратить нейросеть из игрушки в надёжного помощника: разложить ручную работу на шаги, составить запрос, который действительно выполняется, настроить память без утечек и задать нужные тон, формат и уровень качества. Вы научитесь поручать ИИ работу с входящими, документами и календарём, строить развилки для решений, проверять результат и вовремя включать ручной тормоз. Семидневный эксперимент поможет начать без героизма, а главы о командной работе и личной операционной системе — встроить автопилот в повседневную жизнь. Книга создана с помощью ИИ. Обложка: GPT Image - Лицензия

© Тьюрин М., 2026

© Автор, 2026

## Содержание

|                                             |    |
|---------------------------------------------|----|
| Ручная работа под микроскопом               | 5  |
| Запрос, который становится рабочим заданием | 13 |
| Память без утечек                           | 22 |
| Конец ознакомительного фрагмента.           | 27 |

# Марк Тьюрин

## Работа на автопилоте: Как поручить нейросети ежедневные задачи

### Ручная работа под микроскопом

После общего разговора о роли нейросети вопрос становится практичнее: какую именно задачу отдать ей первой? Ответ не найти в списке возможностей модели. Его можно обнаружить, если остановить один обычный рабочий день и рассмотреть не должность целиком, а отдельные действия, из которых она складывается.

В 8:47 у Анны уже было четыре непрочитанных сообщения.

Одно пришло от сотрудника: клиент снова прислал документы не в том формате. Второе — от самого клиента: длинное письмо с одиннадцатью вопросами, вложением и пометкой «срочно». Третье появилось в корпоративном мессенджере от Ильи: «Нужен файл за прошлую неделю. Кажется, у нас две версии». Четвёртое отправила Марина, операционный директор: к обеду подготовить короткую сводку по обращениям с просроченным ответом.

Анна открыла письмо клиента, пролистав его до середины и отправилась искать карточку обращения. В рабочем хранилище нашлись три похожих файла: «Итоги\_май», «Итоги\_май\_новый» и «Итоги\_май\_финал». В последнем не оказалось нужной вкладки. Через несколько минут выяснилось, что данные обновлялись в другом документе.

К 9:03 Анна уже не искала информацию, а приводила в порядок таблицу. В одном столбце даты были записаны как «12.05», в другом — как «12 мая 2024», а в третьей часть значений отображалась числами. Потом пришлось вручную убрать лишние пробелы, переименовать заголовки и проверить, не исчезли ли строки после копирования.

В 9:19 позвонила руководитель группы. У клиента возникла претензия к начислению. Нужно было не просто найти письмо, а восстановить контекст: что ему уже обещали, кто отвечал и можно ли предложить исправление без дополнительного согласования.

В 9:27 написал Павел, непосредственный руководитель Анны: «К 11:00 нужна повестка на встречу с руководителями групп. Возьми открытые вопросы, просрочки и решения с прошлой недели».

Через семь минут Илья прислал ещё одно сообщение: «Если сегодня не соберём сводку, я опять буду переносить цифры вручную до вечера».

Анна открыла чистый лист и записала шесть глаголов: ответить, найти, привести, свести, подготовить, проверить. На первый взгляд всё это выглядело как одно большое поле для автоматизации. Можно было сказать: «Нейросеть, разгрузи мой день». Но такая формулировка скрывала главное: за каждым глаголом стояли разные действия, разные источники информации и разная цена ошибки.

#### **Обычное утро, в котором нет одной задачи**

Если автоматизировать должность целиком, получится непрозрачный комок поручений. Стоит разобрать рабочий день на отдельные операции — и появляются границы.

Анна разделила свои действия на пять зон.

Входящие — письма, сообщения, файлы, звонки, заявки и уведомления, которые нужно принять, прочитать, классифицировать или направить дальше.

Поиск — нахождение нужного файла, факта, фрагмента переписки, пункта инструкции или прежней договорённости.

Преобразование — приведение материала к нужному виду: очистка таблицы, выделение полей, перевод свободного текста в список, сравнение двух версий или группировка обращений.

Создание — подготовка нового результата на основе исходных данных: черновика ответа, сводки, повестки, отчёта или списка поручений.

Контроль — проверка сроков, пропусков, расхождений, обязательных полей и соответствия правилам.

Эти зоны не совпадают с названиями должностей. Подготовка еженедельной сводки, например, включает поиск файлов, преобразование таблиц, создание текста и контроль итоговых значений. Пока такая задача называется просто «сделать отчёт», непонятно, что именно можно передать нейросети, а что останется ответственностью Анны.

Павел попросил её объяснить, что именно она собирается автоматизировать.

«Входящие, ответы, сводки, таблицы и повестки», — сказала Анна.

**«Это названия результатов, а не действий. Что исчезнет из твоего дня?»**

Анна открыла список.

«Ручная сортировка писем. Перенос данных из нескольких файлов. Написание типовых ответов. Составление итогов встреч. Сбор тем для повестки».

**«Кто решит, что письмо срочное?»**

«Пока я».

**«Кто решит, можно ли клиенту обещать компенсацию?»**

«Тоже я».

**«Кто проверит, что в сводке не появилась выдуманная цифра?»**

«Я или Илья».

**Павел кивнул:**

«Значит, речь не о передаче работы целиком. Речь о передаче отдельных операций с понятной точкой контроля».

Эта фраза изменила список Анны. Вместо «автоматизировать ответы» она записала: «подготовить черновик типового ответа без отправки». Вместо «отдать нейросети отчёт» появилось: «свести четыре таблицы к единому формату и показать расхождения». А вместо «пусть сама разберёт входящие» — «присвоить письмам категории по заранее заданным правилам».

**Илья, который подключился к разговору позже, обозначил свою границу:**

«Если модель изменит значения, а не только формат, мы получим красивую ошибку. Мне нужен журнал изменений и возможность сравнить исходник с результатом».

**Марина задала другой вопрос:**

«Куда попадут письма клиентов? В них есть фамилии, телефоны, номера заказов и иногда сведения о договорах».

Вопрос о данных возникает ещё до выбора конкретного сервиса. Если в задаче есть персональные данные, нельзя просто скопировать переписку в любой внешний инструмент. Компания должна использовать разрешённый корпоративный контур или предварительно обезличить материал. Требования к обработке персональных данных в России, в том числе предусмотренные 152-ФЗ, не отменяются только потому, что задачу выполняет нейросеть. Если порядок передачи данных не определён, это отдельный риск, а не техническая мелочь.

**Четыре фильтра для подходящей задачи**

Повторяемость сама по себе не делает задачу хорошим кандидатом. Ежедневное решение о компенсации клиенту тоже повторяется, но передавать такое решение модели нельзя. Анна использовала четыре фильтра.

Первый — повторяемость.

Задача должна возникать достаточно часто и проходить по похожей схеме. Если операция встречается раз в квартал и каждый раз требует нового расследования, настройка автопилота может обойтись дороже ручной работы. Если же сотрудник каждую неделю сортирует десятки сообщений, переносит одни и те же поля или собирает сводку по одной структуре, повторяемость налицо.

Важно учитывать не только частоту, но и форму. Двадцать разных писем не обязательно образуют одну задачу. В половине случаев может потребоваться найти информацию, которой нет в системе, а в другой — согласовать исключение. Это уже два разных сценария.

Второй — ясность входа.

Нужно понимать, с чем работает нейросеть: с письмом, таблицей, стенограммой встречи, списком открытых вопросов или карточками обращений. Должно быть ясно, какие данные разрешено использовать и где находится источник.

«Найди нужную информацию где-нибудь в переписке» — слабый вход. «Вот письма за неделю, карточки обращений и список статусов; выдели просроченные случаи» — уже рабочая заготовка.

Если вход нельзя собрать без долгого ручного поиска, сначала нужно автоматизировать поиск или изменить процесс хранения. Модель не исправит отсутствие единого источника. Она лишь быстрее обработает то, что ей дали, включая лишние и устаревшие файлы.

Третий — проверяемость выхода.

Результат должно быть возможно сравнить с источником, правилом или образцом. Проверить формат дат легко. Убедиться, что в повестке есть все обязательные пункты, тоже несложно. Черновик ответа можно сверить с утверждённым шаблоном и фактами из карточки обращения.

Сложнее проверить вывод вроде «клиент настроен агрессивно» или «этот случай можно закрыть». Здесь нет простой границы правильного ответа. Модель способна предложить гипотезу, но такая задача требует человеческого решения и дополнительных источников.

Четвёртый — цена ошибки.

Цена ошибки складывается не только из денег. В неё могут входить потерянное время, повторная работа, пропущенный срок, недовольство клиента, утечка данных или необходимость объяснять руководству, почему клиенту отправили неверное обещание.

Если модель перепутает формат даты во внутренней черновой таблице, ошибку можно заметить при выборочной проверке. Если она отправит клиенту неверный срок возврата, последствия будут иными. Один и тот же уровень точности может быть приемлемым для внутреннего списка и совершенно недостаточным для сообщения, которое уходит наружу.

Задача хорошо подходит для первого эксперимента, когда она часто повторяется, имеет понятный вход, проверяемый выход и низкую либо контролируемую цену ошибки. Если одного критерия не хватает, задачу не обязательно вычёркивать. Её можно перевести в более осторожный режим: использовать только для черновиков, рекомендаций или обязательной проверки каждой строки.

### **Где заканчивается действие и начинается решение**

Самая опасная путаница возникает между действием и решением.

Действие — это преобразование материала по заданным правилам. Решение — выбор, который меняет обязательства, статус, деньги, отношения или дальнейший ход процесса.

Разобрать письмо по темам — действие. Решить, признавать ли претензию обоснованной, — решение.

Найти в карточках все обращения без ответа больше двух дней — действие. Закрыть эти обращения как неактуальные — решение.

Составить черновик сообщения по утверждённому шаблону — действие. Отправить клиенту обещание о сроке или скидке — решение.

Сравнить две версии таблицы и показать расхождения — действие. Выбрать, какая цифра верная, если источники расходятся, — решение.

Нейросеть может подготовить материал для решения, но это не означает, что ей передали право решать. В рабочем контуре нужно прямо зафиксировать, чего модель не делает: не отправляет сообщения, не удаляет записи, не меняет статусы, не утверждает выплаты и не выносит окончательные выводы по спорным случаям.

Та же граница встречается и за пределами работы. Нейросеть может превратить длинную семейную переписку в список покупок и дел к поездке, но не должна самостоятельно выбирать дозировку лекарства для ребёнка. Она может сравнить варианты подарка для друга по цене, сроку и характеристикам, но перевод денег и окончательный выбор остаются за человеком. Она может собрать из группового чата список предложенных мест, но не обязана говорить от имени участника в конфликте.

Автопилот в этой книге — не режим, в котором человек исчезает. Это ограниченный контур: нейросеть получает подготовленный вход, выполняет заранее оговорённое действие и возвращает результат, а человек сохраняет контроль над решением и последствиями.

### **Матрица, которая возвращает контроль**

После четырёх фильтров Анна не стала присваивать задачам единый «процент готовности к автоматизации». Такой показатель создавал бы ложную точность. Вместо этого она использовала матрицу с двумя вопросами: насколько легко проверить результат и насколько дорого будет ошибиться.

Если результат легко сверить, а ошибка дёшева, задача попадает в зону первого делегирования. Туда относятся приведение таблиц к единому формату, выделение дат и сумм, подготовка структуры повестки, извлечение поручений из заметок. Проверка всё равно нужна, но её можно проводить выборочно или сосредоточиться на ключевых полях.

Когда результат проверяем, но цена ошибки заметна, необходим обязательный человеческий контроль. Так можно работать с сортировкой входящих, черновиками клиентских ответов и сводками встреч. Модель ускоряет подготовку, но готовый результат просматривается до того, как он повлияет на клиента или команду.

Если результат трудно проверить или ошибка затрагивает деньги, права, персональные данные, репутацию и обязательства компании, решение нельзя передавать модели. Нейросеть может собрать факты, выделить варианты и указать на расхождения, но окончательный вывод делает сотрудник с соответствующими полномочиями.

Из этого следуют несколько простых правил.

Если модель только меняет форму материала, а не его смысл, это кандидат на ранний запуск.

Если результатом становится внешнее сообщение, изменение статуса, платёж, предоставление доступа, увольнение, компенсация или юридически значимое обещание, на последнем шаге должен оставаться человек.

Если заранее невозможно описать, как отличить правильный результат от неправильного, сначала нужно создать критерии проверки.

Если в задаче используются персональные данные, ещё до технического эксперимента необходимо определить разрешённый контур и состав передаваемых данных.

Если модель должна сама решать, что считать исключением, сначала следует ограничить область применения правил, а спорные случаи оставить человеку.

Собственная оценка модели, даже если она говорит о «высокой уверенности», такой проверки не заменяет. Уверенный тон не является доказательством.

### **Пять кандидатов Анны**

Анна выбрала пять задач не потому, что они казались самыми эффектными, а потому, что для каждой можно было описать вход, выход и контроль. Это были кандидаты на автопилот, а не обещание полной автоматизации отдела.

### **Сортировка входящих**

До эксперимента Анна открывала сообщения по одному и вручную решала, что делать дальше. На письмо уходило в среднем две-три минуты: прочитать его, найти контекст, определить срочность, переслать сотруднику или поставить напоминание.

Она ввела пять категорий: «ответить сегодня», «передать сотруднику», «запросить данные», «жалоба или риск», «не требует действия». Нейросеть должна была присвоить сообщению категорию, кратко объяснить причину и указать предполагаемый срок реакции. При этом ей запрещалось удалять сообщения, отправлять ответы или менять статус обращения.

### **Рабочая реплика для такого контура могла звучать так:**

«Распредели входящие по пяти категориям. Используй только текст сообщения и указанные в нём факты. Для каждой записи укажи категорию, причину и необходимое действие. Не отвечай отправителю, не закрывай обращение и отдельно вынеси случаи, в которых не хватает данных».

Все сообщения с признаками жалобы, просрочки, финансового спора или обещания со стороны компании требовали обязательной проверки. Остальные категории можно было проверять выборочно, но в первые дни Анна просматривала весь результат, чтобы понять, где модель ошибается.

Анна посчитала: 80 сообщений в неделю по 2,5 минуты на ручную обработку — 200 минут. Проверка результата после сортировки займёт около 35 минут. Исправление ошибок она оценила в 15 минут в неделю, исходя из недавних случаев неверной маршрутизации. Предварительная экономия составила 200 минус 35 и 15, то есть около 150 минут в неделю.

Но в эту цифру не входила цена пропущенного важного письма. Поэтому задача получила статус «только с контролем». Её нельзя было запускать на удаление сообщений, закрытие обращений или автоматический ответ.

### **Подготовка черновиков типовых ответов**

Вторая задача возникала около 30 раз в неделю. Клиенты спрашивали о статусе обращения, перечне документов, сроке обработки и следующем шаге. На ручной ответ уходило в среднем восемь минут: открыть карточку, найти шаблон, проверить данные, сформулировать текст и перечитать его.

Нейросети можно было поручить подготовить черновик, но только по утверждённым шаблонам. В запросе следовало зафиксировать ограничения: не придумывать сроки, скидки, компенсации и причины задержки; не заполнять пропуски догадками; не отправлять сообщение.

### **Пример рабочего задания:**

«Подготовь черновик ответа по утверждённому шаблону. Используй только факты из письма и карточки обращения. Если данных недостаточно, поставь пометку “нужно уточнить”, не додумывай. Не обещай сроков, скидок или компенсаций, которых нет в исходных данных. В конце отдельно перечисли места, требующие проверки. Черновик не отправлять».

Анна проверяла три вещи: отвечает ли текст на вопрос клиента, совпадают ли факты с карточкой и нет ли в нём фразы, которая создаёт новое обязательство. Любой ответ по жалобе, компенсации, спорному начислению или договорным условиям она готовила вручную.

Расчёт выглядел так: 30 черновиков умножить на 8 минут — 240 минут ручной работы. Проверка занимала около 45 минут, исправление ошибок и повторный поиск фактов — ещё примерно 20 минут в неделю. Предварительная экономия составляла 175 минут. При этом итоговая отправка в любом случае оставалась за сотрудником.

### **Сводка встречи**

После каждой встречи Анна или Илья вручную восстанавливали, что обсуждали, какие решения приняли, кто за что отвечает и к какому сроку. На одну встречу уходило около 35 минут. В неделю таких встреч было четыре.

Подходящим результатом считалась не «красивая стенограмма», а короткая рабочая сводка из пяти частей: принятое решение, поручение, ответственный, срок, открытый вопрос. Если в записи звучало несколько вариантов и окончательного решения не было, это следовало обозначить как обсуждение, а не превращать в утверждённый пункт.

**Нейросети можно было дать заметки или расшифровку встречи и попросить:**

«Выдели только решения, поручения, сроки и вопросы без ответа. Для каждого поручения укажи источник в заметках. Не называй решением предложение, если в тексте нет подтверждения. Если ответственный или срок не названы, укажи это отдельно».

Человек проверяет имена, даты и смысл спорных фрагментов. Модель не должна самостоятельно решать, что именно было согласовано. При наличии в записи персональных данных или коммерческой информации нужно использовать разрешённое хранилище и не передавать лишний материал.

Ручная работа занимала 140 минут в неделю. Проверка подготовленных сводок — около 20 минут, исправление неточностей — ещё 10. Предварительная экономия составляла 110 минут. Выгода была заметной, но контроль решений и сроков оставался обязательным.

**Приведение таблиц к единому формату**

Илья тратил на эту работу почти половину недели. Ему приходили файлы от разных руководителей: с разными названиями столбцов, форматами дат, разделителями, пустыми значениями и дублирующимися строками. Само приведение к единому виду занимало около 30 минут на файл. Таких файлов набиралось четыре в неделю.

Это оказался самым чистым кандидатом. Нейросети можно было поручить переименовать столбцы, привести даты к единому формату, убрать лишние пробелы, отметить пустые значения и показать дубли. Но ей нельзя было молча менять суммы, удалять строки или выбирать правильное значение при расхождении источников.

**Правило для задачи выглядело так:**

«Приведи файлы к единой структуре. Меняй только названия полей, формат дат, пробелы и очевидные технические дубли. Не изменяй суммы и текстовые значения. Сохрани список всех изменений. Отдельно перечисли строки, в которых источники расходятся».

Илья проверял количество строк до и после обработки, контрольные суммы, несколько случайных записей и все строки, отмеченные как спорные. Если цифры в двух файлах различались, модель могла показать расхождение, но не имела права выбрать итоговую цифру.

Расчёт был таким: 4 файла умножить на 30 минут — 120 минут. Проверка занимала около 25 минут, исправление ошибок — 10. Предварительная экономия составляла 85 минут. Эта задача попала в зону первого делегирования, но только в режиме копии: исходные файлы сохранялись, а результат не заменял источник.

**Подготовка повестки**

Повестка на встречу возникала три раза в неделю. Анна просматривала прошлые протоколы, список открытых вопросов, просроченные поручения и новые запросы руководителей. На сбор материалов уходило около 20 минут.

Нейросеть могла предложить структуру: тему, цель обсуждения, необходимые материалы, участников и ожидаемый результат. На вход ей подавались предыдущая сводка, список нерешённых вопросов и новые сообщения. Анна сама решала, какие темы действительно требуют встречи, что перенести и какие вопросы поставить первыми.

**Рабочее задание можно было сформулировать так:**

«Собери проект повестки по открытым вопросам, просроченным поручениям и новым запросам. Для каждого пункта укажи цель обсуждения и нужные материалы. Не добавляй

темы, которых нет в исходных данных. Не расставляй приоритеты окончательно, а предложи порядок и пометь спорные пункты».

60 минут ручной работы в неделю превращались в 15 минут проверки и около 5 минут исправлений. Предварительная экономия составляла 40 минут. Задача была небольшой по объёму, но полезной: она снижала количество мелких решений в начале дня.

Так Анна получила пять кандидатов.

Сортировка входящих: 150 минут предварительной экономии, обязательная проверка важных сообщений.

Черновики типовых ответов: 175 минут, проверка каждого текста до отправки.

Сводки встреч: 110 минут, проверка решений, имён и сроков.

Приведение таблиц к единому формату: 85 минут, сравнение исходника и результата.

Подготовка повестки: 40 минут, финальное решение Анны по приоритетам.

Суммарная ручная работа составляла 760 минут в неделю. Проверка занимала 140 минут, ожидаемое исправление ошибок — около 60 минут. Грубая оценка давала 560 минут, или чуть больше девяти часов. Это не обещание и не прогноз результата, а верхняя граница при сохранении прежнего объёма задач и условии, что настройка уже выполнена.

**Павел посмотрел на расчёт и спросил:**

**«Ты действительно ждёшь девять свободных часов?»**

«Нет. Я получила девять часов теоретической экономии до запуска. В первую неделю нужно вычестть настройку, обучение сотрудников, разбор ошибок и дополнительную проверку. Для планирования я возьму половину расчётной цифры, а потом заменю оценку фактическими данными».

**«С чего начнёшь?»**

«С таблиц и повестки. Они безопаснее. Затем добавлю сводки. Сортировку входящих и черновики ответов подключу только после того, как проверю правила и работу с персональными данными».

Это решение было менее эффективным, чем обещание «автоматизировать отдел за один день». Зато у него были границы. Анна знала, что именно измерит и где остановит эксперимент.

**Проверка перед первым запуском**

Не нужно ждать, пока накопится идеальный список всех рабочих процессов. Достаточно взять один вчерашний день и разобрать его на десять-пятнадцать отдельных операций.

Не записывайте «подготовить отчёт». Зафиксируйте конкретные шаги: найти четыре файла, проверить даты, убрать технические пробелы, собрать одинаковые поля, сравнить итоговые суммы, написать выводы. Не записывайте «ответить клиенту». Разложите задачу иначе: прочитать письмо, найти карточку, проверить статус, выбрать шаблон, написать черновик, отправить его после проверки.

Для каждой операции заполните короткую карту.

Задача: что именно повторяется.

Вход: какие письма, файлы, записи или правила нужны.

Действие: что должно произойти с входными данными.

Выход: какой результат должен появиться.

Проверка: по чему человек поймёт, что результат правильный.

Цена ошибки: что произойдёт при неверном результате.

Частота и время: сколько раз в неделю возникает задача и сколько минут занимает вручную.

Затем проверьте сам процесс.

Повторяется ли задача по похожей схеме, а не только носит одно и то же название?

Известен ли источник входных данных и есть ли к нему доступ?

Можно ли передать нейросети только необходимый объём данных?

Существует ли способ сверить результат с образцом, источником или правилами?

Остаётся ли человек на последнем шаге там, где возникает внешнее обязательство?

Записаны ли заранее запрещённые действия: отправка, удаление, изменение статуса, финансовое решение?

Есть ли исходная цифра для сравнения: ручное время, число ошибок и объём проверки?

Если на любой вопрос нет ответа, это не означает провал автоматизации. Скорее, процесс ещё недостаточно определён. Для первого запуска достаточно выбрать одну-две задачи из пяти, а не пытаться охватить весь список.

Анна начала с таблиц. Через неделю она сравнила расчёт с фактом: часть времени действительно исчезла, но появились новые минуты на проверку спорных строк. Затем она добавила сводки встреч и только после этого вернулась к сортировке входящих. Каждый следующий шаг опирался не на впечатление от «умной» модели, а на карту входов, выходов, проверок и ошибок.

Теперь задачу удалось назвать достаточно точно, чтобы передать её нейросети. Но название ещё не стало рабочим заданием. Нужно было описать исходные данные, формат результата, ограничения и критерии приёмки. С этого начинается следующий шаг: запрос, который превращает намерение «помоги с письмами» в понятную операцию с измеримым выходом.

## Запрос, который становится рабочим заданием

Вы уже умеете выбирать задачи для автопилота по четырём признакам: понятны входные данные, результат, способ проверки и цена ошибки. Но даже подходящая задача может рассыпаться при передаче нейросети. Человек достраивает недостающий смысл по ситуации, а модель видит только то, что попало в запрос и приложенные материалы.

Возьмём одно и то же письмо клиенту.

«Добрый день! По счёту №417 от 10 июня в нашей системе пока не отображается оплата. Отгрузка заказа запланирована на 18 июня. Если платёж уже отправлен, направьте копию платёжного поручения, чтобы мы проверили его у банка. Если оплаты ещё не было, сообщите, когда её ожидать. После подтверждения оплаты мы уточним статус отгрузки».

К этому письму можно добавить короткую команду: «Сделай покороче».

**Результат может выглядеть так:**

«Добрый день! Оплата по счёту №417 не поступила. Пришлите платёжное поручение или сообщите срок оплаты».

**Или так:**

«Коллеги, отгрузка 18 июня под угрозой из-за неоплаченного счёта. Срочно подтвердите оплату».

**Или ещё жёстче:**

«Счёт №417 не оплачен. Отгрузка не состоится до подтверждения оплаты».

Все три варианта выглядят правдоподобно, но решают разные задачи. Первый просит клиента выполнить одно из двух действий. Второй усиливает срочность и добавляет оценку «под угрозой». Третий превращает запланированную отгрузку в отменённую, хотя исходный текст этого не утверждает. Модель не ошиблась случайно: ей не сообщили, что именно нужно сохранить, что можно изменить и по каким признакам считать результат готовым.

Запрос — не вежливая просьба к нейросети и не магическая формула. Это компактное рабочее задание. В нём должны быть обозначены цель, материалы, адресат, правила обработки и форма результата. Если в исходных данных есть пробел, запрос должен не заставлять модель угадывать, а направлять её к уточняющему вопросу.

**Мифы, из-за которых запросы не работают**

Миф первый: чем короче запрос, тем лучше.

Короткая команда удобна человеку, который уже держит весь контекст в голове. Для модели она часто означает свободу выбора. Команда «сделай понятнее» не говорит, что именно непонятно: терминология, порядок фактов, длина предложений или логика действия. Команда «перепиши для клиента» не уточняет, знает ли клиент историю вопроса или видит письмо впервые.

Коротким может быть хороший запрос, если задача повторяется, данные стандартизованы, а правила уже закреплены в шаблоне. Но сокращать нужно не условия, а пояснения, без которых смысл не теряется. Фраза «убери обвинительный тон, сохрани даты и попроси подтвердить один из двух вариантов» полезнее, чем длинное описание всей истории отношений с клиентом.

Миф второй: модель сама поймёт контекст из приложенного текста.

Исходный материал и цель работы — разные вещи. Из стенограммы встречи можно сделать протокол, список поручений, письмо участникам или короткую записку руководителю. Сам текст встречи не определяет, какой результат нужен. Контекст — это не весь архив по теме, а только сведения, необходимые для принятия решения и подготовки ответа.

Миф третий: достаточно назначить роль.

Фраза «Ты опытный редактор» задаёт общий угол зрения, но не устанавливает критерий качества. Редактор может сделать письмо дружелюбным, строгим, юридически осторожным или предельно коротким. Роль становится полезной, когда связана с конкретной функцией: «Ты — редактор клиентской переписки. Твоя задача — сохранить подтверждённые факты, убрать обвинения и привести письмо к одному понятному действию».

Миф четвёртый: неудачный результат нужно просто попросить переделать.

Команда «попробуй ещё раз» иногда меняет формулировки, но не устраняет причину сбоя. Если модель придумала срок, нужно запретить добавлять неподтверждённые даты. Если она сделала текст слишком жёстким, следует описать адресата и допустимый тон. Если потеряла важное условие, его надо вынести в ограничения или критерий готовности.

Миф пятый: удачный пример не нужен, достаточно описать тон словами.

Слова «деловой», «живой» и «дружелюбный» допускают слишком много трактовок. Один короткий одобренный образец одновременно показывает длину, порядок аргументов, способ обращения и плотность текста. Но образец должен задавать форму, а не заменять исходные данные. Факты из старого письма нельзя переносить в новое только потому, что они присутствуют в примере.

### **Конструкция рабочего запроса**

У рабочего запроса есть понятная последовательность. Сначала определяется цель, затем материалы, адресат, ограничения и форма выдачи. Роль задаёт рабочую позицию модели. Перед финальным ответом включается фильтр предположений, а критерий готовности помогает проверить результат.

#### **Роль**

Роль — это рабочая позиция модели в конкретной задаче. Она не должна звучать как украшение. Сравните:

«Ты — профессионал высокого уровня».

«Ты — редактор деловой переписки. Сохраняй подтверждённые факты и не усиливай утверждения».

Во втором варианте есть функция и граница ответственности. Для сводки встречи подойдёт роль аналитика протоколов, для обработки обращений клиентов — классификатора входящих сообщений, для повестки — координатора рабочих совещаний.

Роль не наделяет модель правом принимать решения за пользователя. Фраза «Ты — руководитель проекта» не означает, что модель может назначить ответственных или утвердить дату. Если такого права нет в исходных данных, она должна обозначить пробел.

#### **Цель**

Цель отвечает на вопрос: зачем нужен результат? Фраза «Сделай покороче» описывает действие над текстом, но не объясняет, какого эффекта нужно добиться.

Рабочая формулировка может выглядеть так: «Сократи письмо до пяти предложений, чтобы клиент понял текущий статус оплаты и выполнил одно из двух действий: прислал подтверждение платежа или сообщил ожидаемую дату оплаты».

Цель лучше связывать с решением или действием. Для сводки это может быть «дать руководителю список принятых решений и незакрытых вопросов». Для классификации — «направить каждое обращение в правильную рабочую очередь». Для повестки — «за 45 минут подготовить участников к выбору даты запуска».

Проверить цель можно простым вопросом: если убрать весь исходный текст, сможете ли вы одним предложением объяснить, зачем нужен итог? Если нет, модель тоже будет выбирать цель сама.

### **Исходные данные**

Здесь нужно зафиксировать, на каких материалах разрешено работать. Это может быть письмо, стенограмма, список обращений, календарные ограничения, утверждённая классификация или несколько файлов с разным статусом.

**Полезно обозначать границы источника прямо:**

«Используй только текст обращения ниже».

«Даты бери из блока “Подтверждено”».

«Не используй сведения из старой версии инструкции».

«Если в материалах есть противоречие, остановись и укажи на него».

Фраза «проанализируй всё» плохо задаёт границы. Непонятно, какие документы считать главными, какой период нужен и можно ли привлекать знания извне. Для повторяющейся задачи лучше вставлять материалы в одинаковом порядке: сначала исходный текст, затем справочник категорий и после него дополнительные условия.

**Адресат**

Адресат — тот, кто будет читать или использовать результат. У него есть определённый уровень знаний, интерес, полномочия и ожидаемое действие.

Письмо клиенту должно объяснить статус без внутренней терминологии. Сводка для руководителя должна отделить решения от подробностей обсуждения. Список для оператора поддержки обязан показывать, куда направить обращение. Повестка для участников должна объяснять, к какому решению им нужно прийти.

Адресат — это не только должность. Нужно учитывать отношения и ситуацию: это первое обращение или продолжение переписки, видел ли получатель номер заказа, требуется ли от него действие сегодня или достаточно информирования. Если эти сведения неизвестны, лучше выбрать нейтральную формулировку и явно обозначить ограничение.

**Ограничения**

Ограничения определяют, что нельзя потерять и что нельзя добавлять. Здесь задаются длина, тон, допустимые утверждения, обязательные элементы и запрещённые действия.

**Например:**

«Не называй оплату просроченной: в системе пока отсутствует только подтверждение».

«Не меняй даты и номера документов».

«Не обещай клиенту новую дату отгрузки».

«Не используй обвинительные формулировки».

«В каждой строке классификации указывай только одну основную категорию».

Ограничение должно быть проверяемым. Фраза «сделай красиво» не проверяется. А условие «не более 600 знаков, одна просьба о действии, без восклицательных знаков и без новых фактов» вполне можно проверить.

**Форма выдачи**

Форма отвечает на вопрос: как должен выглядеть готовый результат? Для письма это тема и основной текст. Для протокола — решения, поручения, сроки и незакрытые вопросы. Для классификации — строка на каждое обращение, категория, основание и уровень уверенности. Для повестки — время, тема, ожидаемый результат и необходимые материалы.

Если форму не задать, модель может выдать эссе вместо списка, сплошной текст вместо рабочих полей или длинное объяснение вместо готового письма. Даже простая инструкция «сначала итог, затем список вопросов» заметно повышает пригодность результата.

К этой конструкции добавляются ещё два элемента.

**Проверка предположений**

Перед выполнением задачи модель должна отделить подтверждённые факты от предположений и неизвестных данных.

**Факт** — это информация, которая прямо указана в источнике или подтверждена пользователем. В исходном письме фактами могут быть номер счёта, дата документа, отсутствие отметки об оплате в системе и запланированная дата отгрузки.

**Предположение** — трактовка, которая кажется вероятной, но не следует из материала напрямую. Например, отсутствие отметки об оплате может быть связано с ошибкой синхронизации, а не с отсутствием платежа.

**Неизвестное** — обстоятельство, для которого в материалах недостаточно данных. Например, неизвестно, отправил ли клиент платёж, кто должен согласовать итог встречи или можно ли перенести совещание.

#### **В рабочий запрос полезно вставлять такой блок:**

«Перед подготовкой ответа раздели данные на три части: подтверждено, предполагается, неизвестно. Если неизвестное влияет на сумму, срок, обещание, адресата, причину проблемы или назначение ответственного, задай уточняющий вопрос и не выдавай финальный вариант. Если пробел не влияет на смысл, продолжи работу, но пометь использованное предположение».

Такой фильтр не заставляет модель задавать вопросы по каждой мелочи. Неизвестная форма приветствия обычно не блокирует письмо: можно выбрать нейтральное обращение. А неизвестность в вопросе о том, был ли отправлен платёж, не позволяет утверждать, что счёт не оплачен.

#### **Критерий готовности**

Критерий готовности превращает субъективное «нормально» в короткую проверку. Для письма он может звучать так: «Текст готов, если клиент понимает текущий статус, видит одно понятное действие, не получает неподтверждённых обещаний, а все даты и номера совпадают с источником».

Критерий не заменяет проверку человеком. Он заранее говорит, что именно нужно проверить. При высокой цене ошибки финальный текст всё равно должен просмотреть ответственный сотрудник.

#### **Лабораторное упражнение**

Возьмите не идеальную учебную задачу, а один недавний реальный пример: письмо, протокол, набор обращений или черновик повестки. Скопируйте исходную команду без правок. Если там написано «сделай понятнее», оставьте именно эту фразу.

Затем разберите задачу по семи шагам.

Сначала назовите действие, которое адресат должен выполнить после получения результата.

Затем выпишите только те материалы, на которые модель имеет право опираться.

После этого разделите сведения на факты, предположения и неизвестные.

Далее опишите адресата не одним словом, а через его задачу: что он знает, чего не знает и что должен сделать.

Следом укажите ограничения, которые нельзя нарушить.

Потом задайте форму выдачи.

И наконец, сформулируйте условие «готово, если».

Если после этого запрос стал длиннее, это не проблема. Его задача — сэкономить время на исправлениях и уменьшить число угадываний. После нескольких повторений часть полей можно объединить в постоянный шаблон.

#### **Четыре задачи через один шаблон**

##### **Письмо клиенту**

Внешняя переписка особенно чувствительна к предположениям. Внутреннее сообщение можно быстро поправить, а письмо клиенту создаёт ожидание, которое потом придётся выполнять.

Исходные данные: по счёту №417 от 10 июня в системе пока не отображается оплата; отгрузка запланирована на 18 июня; если платёж уже отправлен, клиент может прислать копию платёжного поручения; если оплаты ещё не было, клиент может сообщить ожидаемую дату.

Факты здесь таковы: номер и дата счёта, отсутствие отметки в системе, запланированная дата отгрузки и два возможных действия клиента.

Предположения: клиент уже знает, что отгрузка запланирована; отсутствие отметки связано именно с банковской обработкой; письмо попадёт сотруднику, который отвечает за оплату.

Остаётся неизвестным, отправил ли клиент платёж, актуальна ли дата отгрузки, нужно ли копировать в письмо менеджера по продажам и насколько формальным должно быть обращение.

### **Рабочий запрос:**

Роль: ты — редактор деловой переписки.

Цель: подготовить короткое письмо, которое позволит получить от клиента подтверждение статуса оплаты и не создаст обещаний по отгрузке.

Материалы: используй только исходный текст ниже. Сохрани номер счёта, дату счёта и запланированную дату отгрузки.

Адресат: представитель компании-клиента, знакомый с заказом, но, возможно, не видевший предыдущую переписку.

Ограничения: не называй счёт неоплаченным, потому что подтверждено только отсутствие отметки в системе. Не обвиняй клиента. Не добавляй новую дату отгрузки и не обещай, что заказ будет отгружен. Предложи два варианта действия: прислать платёжное поручение или сообщить ожидаемую дату оплаты. Объём — до пяти предложений.

Форма: сначала тема письма, затем готовый текст. В конце отдельно укажи, какие данные требуют проверки перед отправкой.

Проверка предположений: если для письма необходимы неизвестные сведения, задай вопрос до подготовки финального варианта. Если этих сведений достаточно для нейтрального текста, не останавливайся из-за неизвестной должности получателя.

Критерий готовности: в письме сохранены все даты и номер счёта, нет новых фактов, клиент понимает следующее действие, а формулировка не превращает отсутствие отметки в доказательство отсутствия оплаты.

Такой запрос не заставляет модель угадывать, что означает «короче». Короткость здесь измеряется количеством предложений, а качество — сохранением фактов и понятным действием.

### **Сводка встречи**

Заметки со встречи часто смешивают решения, предложения и вопросы. Фраза «нужно обновить инструкцию к 20 июня» может быть поручением, предварительной идеей или записью чужого мнения. Если модель превратит каждое «можно сделать» в обязательство, сводка создаст несуществующие договорённости.

Исходные данные: встреча длилась 45 минут; обсуждалось обновление внутренней инструкции; в заметках есть предложение завершить работу к 20 июня; обсуждалась необходимость согласования с ответственным подразделением; ответственный и окончательный срок не указаны.

Факты: дата и продолжительность встречи, её тема, наличие предложения о сроке и наличие вопроса о согласовании.

Предположения: дата 20 июня утверждена; обновление инструкции поручено конкретному сотруднику; все участники согласились с предложением.

Неизвестно, кто владеет задачей, утверждён ли срок, кто согласует документ и какие вопросы закрыты окончательно.

### **Рабочий запрос:**

Роль: ты — аналитик рабочих протоколов.

Цель: превратить заметки в сводку, по которой участники увидят принятые решения, открытые вопросы и поручения без выдуманных назначений.

Материалы: используй только заметки встречи. Не добавляй сведения из общих знаний или старых протоколов.

Адресат: участники встречи и руководитель проекта.

Ограничения: не превращай предложение в решение. Если ответственный или срок не назначены, укажи «не назначен» и вынеси вопрос в список требующих уточнения. Сохраняй различие между тем, что обсуждалось, предлагалось, согласовывалось и отклонялось.

Форма: разделы «Решения», «Поручения», «Открытые вопросы». Для каждого поручения укажи действие, ответственного и срок; при отсутствии данных поставь отметку «не указано».

Проверка предположений: перед финальной сводкой проверь, можно ли считать 20 июня утверждённым сроком. Если нет, сформулируй уточняющий вопрос и не записывай эту дату как обязательство.

Критерий готовности: читатель за две минуты видит, что решено, что только предложено и какие сведения нужно подтвердить.

Здесь модель может подготовить предварительный протокол, не останавливаясь: неизвестные поля достаточно пометить. Но если от сводки требуется официальный список поручений, вопрос об ответственном и сроке становится обязательным до публикации.

### **Классификация обращений**

Классификация нужна не для красивой сортировки, а для маршрутизации. Ошибка в категории может отправить обращение не в ту очередь, увеличить срок ответа и скрыть повторяющуюся проблему.

#### **Представим, что на вход поступают сообщения:**

«В личном кабинете сумма заказа списалась дважды».

«Заказ доставлен, но одного товара внутри нет».

«После смены номера телефона не получается войти в личный кабинет».

Допустимые категории заданы заранее: оплата, доставка, комплектация заказа, доступ к личному кабинету, другое.

Факты: тексты обращений, утверждённый список категорий и необходимость выбрать рабочую очередь.

Предположения: двойное списание означает окончательное списание, отсутствие товара связано с ошибкой склада, а смена номера телефона стала причиной сбоя доступа.

Неизвестно, были ли деньги действительно списаны дважды или одна операция только отображается как обрабатываемая, когда доставили заказ, какой у него номер и насколько срочно требуется ответ.

### **Рабочий запрос:**

Роль: ты — классификатор входящих обращений.

Цель: присвоить каждому сообщению одну основную категорию для передачи в рабочую очередь.

Материалы: используй только текст обращения и список категорий ниже.

Адресат: оператор первой линии, который будет направлять сообщения дальше.

Ограничения: не определяй виновника, срочность и размер ущерба, если этого нет в тексте. Не создавай новые категории. Если сообщение подходит к нескольким категориям, выбери основную по описанному действию клиента и отметь альтернативу. Если данных недостаточно, поставь уровень уверенности «низкий» и укажи, что нужно уточнить.

Форма: одна строка на обращение: номер, категория, короткое основание из текста, уровень уверенности, недостающие сведения. В конце посчитай количество обращений по каждой категории.

Проверка предположений: не выдавай предположение за факт. В формулировке «сумма списалась дважды» сохрани именно утверждение клиента, а не вывод о подтверждённом двойном списании.

Критерий готовности: каждое обращение получило одну из разрешённых категорий, основания можно найти в исходном тексте, а спорные случаи отмечены для проверки оператором.

В этом примере полезно требовать короткое основание. Если модель просто выдаст категорию, сотрудник не увидит, почему она так решила, и не сможет быстро исправить спорную строку.

### **Подготовка повестки**

Повестка часто выглядит как список тем, хотя рабочая повестка должна вести к результату. Пункт «обсудить запуск» не говорит, нужно ли принять решение, собрать сведения или только проинформировать участников.

Исходные данные: встреча длится 45 минут; цель — выбрать один из двух вариантов даты запуска внутреннего процесса; участники представлены тремя ролями: руководитель проекта, финансовый специалист и специалист поддержки; есть материалы с рисками и ограничениями по каждому варианту.

Факты: продолжительность встречи, цель выбора, два варианта, роли участников и наличие материалов.

Предположения: все участники имеют право принять решение; материалы будут прочитаны заранее; риски сопоставлены по одному набору критериев.

Неизвестно, кто ведёт встречу, кто фиксирует решение, какие критерии имеют приоритет и можно ли принять решение без дополнительного согласования.

### **Рабочий запрос:**

Роль: ты — координатор рабочей встречи.

Цель: составить повестку на 45 минут, которая приведёт участников к выбору даты запуска или зафиксирует, почему решение перенесено.

Материалы: используй описание двух вариантов и список рисков. Не добавляй новые критерии и не назначай ответственных без указания в исходных данных.

Адресат: участники, которым нужно подготовиться к принятию решения.

Ограничения: каждый пункт должен иметь ожидаемый результат. Разделяй информирование, обсуждение и принятие решения. Оставь несколько минут на фиксацию итогов. Не ставь в повестку тему без объяснения, зачем она нужна.

Форма: название пункта, длительность, цель пункта, необходимый материал, ожидаемый итог. Отдельно укажи, какие данные нужно подтвердить до встречи.

Проверка предположений: если критерии выбора не заданы, не придумывай их. Сформулируй вопрос о критериях или предложи использовать только те, что есть в приложенных материалах.

Критерий готовности: сумма времени равна 45 минутам, в повестке есть блок принятия решения и блок фиксации итогов, а подготовка участников понятна заранее.

Здесь результат можно проверить арифметически: время должно сходиться. Это хороший пример критерия, который не зависит от впечатления редактора.

### **Как использовать образец**

Образец — не украшение и не повод копировать старый текст. Это сжатая демонстрация структуры. Один удачный пример письма может заменить несколько абзацев о желаемом тоне.

**Например, в библиотеку можно положить такой обезличенный образец:**

**Тема: Статус оплаты по счёту №[номер]**

Добрый день!

По счёту №[номер] оплата пока не отображается в нашей системе. Если платёж уже отправлен, пришлите, пожалуйста, платёжное поручение для проверки. Если оплата ещё не проведена, сообщите ориентировочную дату. После проверки мы вернёмся со статусом заказа.

**С уважением,**

[подпись]

Из него можно извлечь структуру: сначала нейтральное обращение, затем подтверждённый статус без обвинения, после этого два возможных действия и в конце осторожное описание следующего шага. В каждом варианте есть одна просьба, нет эмоционального давления и лишних объяснений.

**В новый запрос стоит вставить не только сам образец, но и правило его использования:**

«Используй пример как эталон длины, последовательности и степени формальности. Не переноси из образца в новый текст ни одного факта, номера, даты или обещания. Если образец противоречит ограничениям текущей задачи, приоритет имеют текущие ограничения».

Обычно одного чистого образца достаточно. Три образца могут быть полезны, если они показывают разные допустимые случаи: первое обращение, напоминание и ответ на претензию. Но противоречивые примеры без объяснения только увеличат неопределённость.

**Как превращать удачный запрос в рабочий актив**

Первый удачный результат ещё не означает, что шаблон готов. Он мог получиться потому, что исходный текст был необычно ясным, адресат очевиден, а все даты случайно совпали с ожиданиями. Чтобы запрос стал инструментом, его нужно проверить на нескольких реальных случаях и хранить вместе с результатами.

Для этого подойдёт карточка запроса.

Название задачи: «Клиентское письмо — запрос статуса оплаты».

Назначение: получить от клиента подтверждение платежа или ожидаемую дату оплаты без обвинительных формулировок.

Роль: редактор деловой переписки.

Версия: 0.1.

Дата создания: [дата].

Последняя дата улучшения: [дата].

Исходный пример: обезличенный текст письма и описание ситуации.

Текст запроса: полная рабочая формулировка со всеми слоями.

Критерий готовности: список проверок, которым должен соответствовать результат.

Ожидаемый результат: пример, прошедший проверку ответственным сотрудником.

Фактический результат: ответ модели на исходном примере.

Замечания проверяющего: что пришлось исправить вручную, какие факты потерялись, какие формулировки оказались слишком жёсткими или мягкими.

Типичные ошибки: модель называет платёж просроченным, переносит дату из образца, добавляет обещание по доставке.

План следующего улучшения: например, добавить запрет на новые даты и обязательную проверку статуса отгрузки.

Если после проверки изменилось одно правило, создавайте новую версию, а не переписывайте старую без следа. Версия 0.1 может быть первым рабочим черновиком, версия 0.2 — результатом исправления одного сбоя, версия 1.0 — шаблоном, который устойчиво прошёл несколько разных случаев. Нумерация не обязана быть сложной. Её задача — показать, какой текст запроса дал конкретный результат.

Улучшать шаблон лучше по одному типу ошибки за раз. Если одновременно поменять роль, ограничения, формат и образец, невозможно понять, что именно помогло. Возьмите три-пять прошлых примеров, проверьте на них новую версию и сравните ответы с прежними. Такой набор старых примеров станет проверочным набором для будущих изменений.

Ошибки удобно разделять по причине.

Добавленный моделью факт обычно указывает на размытые границы исходных данных или отсутствие запрета на выдумывание.

Перепутанный тон говорит о том, что не описан адресат или нет подходящего образца.

Пропущенный важный пункт показывает, что цель и критерий готовности сформулированы недостаточно чётко.

Неудобный результат означает, что не задана форма ответа.

Уверенный ответ при нехватке сведений появляется, когда в запросе нет правила об обязательном уточняющем вопросе.

После каждой проверки сохраняйте не только удачный текст, но и исходную ошибку. Библиотека, в которой лежат одни красивые результаты, не помогает предупреждать повторяющиеся сбои. Рабочая библиотека показывает связь между входом, запросом, результатом и исправлением.

Хранить следует обезличенные примеры. Удаляйте из карточек телефоны, адреса, номера договоров, персональные данные и другие сведения, которые не нужны для воспроизведения структуры задачи. Если факт необходим для проверки, заменяйте его понятным маркером: [номер счёта], [дата], [название проекта]. Полезный шаблон должен передавать способ работы, а не собирать архив чужих данных.

Теперь расплывчатая команда «сделай покороче» превращается в проверяемое задание. Модель получает цель, источник, адресата, правила, форму и инструкцию не угадывать критически важный смысл. А запрос, который дал хороший результат, перестаёт быть случайной находкой: он сохраняется, получает версию и проверяется на новых случаях.

Следующий вопрос возникает сразу после создания такой библиотеки: какие материалы можно повторно передавать модели, где их хранить и как не допустить, чтобы удобный контекст превратился в утечку. Именно этому посвящена следующая глава.

## Память без утечек

В 17:20 сотруднику службы поддержки нужно закрыть десять обращений до конца рабочего дня. Он копирует в нейросеть выгрузку из CRM, несколько писем клиентов, внутреннюю переписку, договор, действующий прайс-лист, старую инструкцию и таблицу с реквизитами. На первый взгляд задача разумна: подготовить черновики ответов. Но вместе с полезными сведениями в обработку уходят телефоны, адреса, номера заказов, фрагменты чужих обращений и документы, не имеющие отношения к текущей задаче.

Так возникает обманчивое ощущение качества: чем больше материалов передано, тем точнее должен быть результат. На практике избыток данных усложняет выбор актуального правила, расширяет круг лиц, которые могут получить доступ к информации, и мешает ответить на простой вопрос: почему нейросеть использовала именно этот факт?

Когда у задачи уже определены цель, вход, выход и критерий готовности, нужно отдельно решить, какие сведения действительно войдут в работу. Рабочий контекст — не архив компании и не память обо всём, что когда-либо происходило. Это дозированный набор данных, необходимых для конкретного результата: целей, терминов, правил, примеров и ограничений. Всё остальное должно остаться за пределами обработки.

### **Контекст не равен памяти модели**

Слово «память» удобно, но может ввести в заблуждение. Нейросеть не должна превращаться в хранилище всех документов, переписок и личных данных. Рабочий контекст формирует пользователь или организация: отбирает материалы, проверяет право доступа, отмечает срок действия и решает, что делать с устаревшими сведениями.

Логика здесь проста. Чтобы подготовить ответ клиенту по гарантийному случаю, нейросети может понадобиться следующее.

Цель — составить черновик ответа без обещаний, которых нет в правилах.

Термины — что в компании означают «диагностика», «замена», «ремонт» и «гарантийный случай».

Правила — в каких ситуациях обращение передаётся инженеру или руководителю.

**Пример** — один или два одобренных ответа на похожие обращения.

Ограничения — не указывать непроверенные сроки, не раскрывать внутренние причины отказа и не запрашивать у клиента лишние документы.

Для этого не нужны паспортные данные клиента, полная история его покупок, внутренний чат отдела и все версии прайс-листа за последние два года. Если передать эти сведения «на всякий случай», система получит лишний объём, а организация — лишний риск.

Полезный тест звучит так: если удалить этот фрагмент, сможет ли нейросеть выполнить задачу с тем же качеством? Если да, фрагмент не относится к минимальному контексту. Если нет, нужно понять, какую именно функцию он выполняет: помогает разобраться в термине, подтверждает правило, задаёт формат или содержит факт текущего дела.

### **Сигнал: в пакет попало всё, что нашлось по теме**

Это не признак тщательной подготовки, а отсутствие границы задачи. Особенно опасны папки с названиями «Важное», «Клиенты», «Проекты» и выгрузки из рабочих систем без фильтрации. В них почти всегда смешаны материалы с разными сроками действия, уровнями доступа и назначением.

Что делать. Сначала зафиксировать результат, затем перечислить минимальные факты, без которых он невозможен. Для черновика ответа по задержке доставки могут понадобиться код заказа, текущий статус отправления, утверждённое окно доставки и правило компенсации. Имя клиента, его телефон и полный адрес при этом могут не потребоваться.

**Сигнал: одно правило повторяется в трёх документах и сформулировано по-разному**

Это риск не только ошибки, но и незаметного выбора устаревшей версии. Нейросеть может найти более подробную инструкцию, однако подробность сама по себе не делает документ действующим.

Что делать. Выбрать контролируемый источник, указать дату проверки, а остальные документы явно пометить как архивные или вспомогательные. В рабочем запросе полезно закрепить правило: «Используй только сведения со статусом “действует”; при конфликте покажи расхождение и не выбирай значение самостоятельно».

### **Три слоя рабочего контекста**

Рабочий контекст удобно разделять на три слоя: постоянные сведения, временные материалы и разовые вводные. Это не обязательно три разные функции в используемом сервисе. Скорее, организационная схема, которая помогает не смешивать устойчивые правила с материалами конкретного случая.

#### **Постоянные сведения**

Постоянный слой включает то, что используется во многих однотипных задачах и меняется относительно редко. К нему могут относиться:

- назначение продукта или отдела;**
- словарь внутренних терминов;**
- описание ролей и зон ответственности;**
- утверждённые правила обработки обращений;**
- перечень запрещённых обещаний и формулировок;**
- порядок эскалации сложных случаев;**
- структура типового отчёта;**

ограничения на раскрытие коммерческой информации.

Постоянный не значит вечный. Регламент может измениться после решения руководителя, продукт — после обновления, а порядок эскалации — после перераспределения ответственности. Поэтому у каждого элемента постоянного слоя должны быть дата последней проверки и владелец обновления.

Практическая ошибка возникает, когда в постоянный слой помещают то, что действует только до определённой даты. Например, специальный порядок возврата товара на время акции нельзя хранить рядом с базовой политикой возврата без явной пометки срока. Через месяц нейросеть может применить временное исключение как обычное правило.

#### **Временные материалы**

**Временный слой связан с текущим периодом, проектом или событием. Это могут быть:**

- прайс-лист на текущий месяц;**
- условия действующей акции;**
- расписание поставок на неделю;**
- актуальный список доступных услуг;**
- сроки проекта;**
- статус технического сбоя;**
- новая версия договора до её окончательного утверждения;**
- перечень сотрудников, назначенных на дежурство в конкретный период.**

Для каждого временного материала обязательны дата начала действия и, если возможно, дата окончания. Если срок окончания неизвестен, нужно указать событие, после которого материал следует проверить: публикацию нового прайса, завершение акции, закрытие инцидента или подписание договора.

Именно временные материалы чаще всего становятся источником незаметных ошибок. Цена в таблице может быть правильной на дату создания файла, но уже не действовать сегодня. Срок поставки в письме мог относиться только к одной партии. Реквизиты из старого договора могут попасть в новый документ просто потому, что нейросеть получила обе версии одновременно.

### **Разовые вводные**

Разовые вводные относятся к конкретной задаче. Это текст текущего обращения, несколько строк из заявки, исходные цифры отчёта, один договор на проверку, перечень позиций для сравнения или конкретное письмо контрагента.

Разовая вводная не должна автоматически превращаться в постоянное знание. Если один клиент сообщил об необычном условии, это ещё не новое правило компании. Если в одном проекте применили исключение, его нельзя переносить на другие проекты без подтверждения.

На практике три слоя могут выглядеть так.

В службе поддержки постоянный слой содержит правила гарантии и список случаев для передачи инженеру. Временный — сведения о текущей задержке на складе и действующий срок доставки. Разовый — содержание одного обращения и код конкретного заказа.

В отделе персонала постоянный слой может включать порядок оформления отпуска и правила маршрутизации заявлений. Временный — производственный календарь, график отпусков отдела и действующие локальные формы. Разовый — конкретное заявление без лишних сведений о сотруднике. Медицинские подробности, семейные обстоятельства и копии документов не относятся к контексту, если для результата нужны только дата отсутствия и вид заявления.

В закупках постоянный слой задаёт критерии сравнения предложений и правила согласования. Временный — актуальные условия поставки и бюджет проекта. Разовый — три предложения на конкретную партию товара. Банковские реквизиты, подписи и документы, подтверждающие личность представителей, не нужны для простой таблицы сравнения цен и сроков.

Если слои смешаны, появляется простой сигнал: один и тот же фрагмент приходится каждый раз вручную объяснять нейросети. Значит, материалы не разделены по сроку действия и назначению. Исправление начинается не с более длинного запроса, а с разметки самого контекста.

### **Обезличивание до передачи в обработку**

Обезличивание — это удаление или замена сведений, по которым человека можно определить прямо или косвенно. Просто стереть имя недостаточно. Сочетание даты, редкой должности, небольшого населённого пункта, суммы и обстоятельств иногда позволяет восстановить личность даже без фамилии.

Перед обработкой данные нужно разделить на три группы.

Первая группа необходима для результата. Например, при анализе обращений важны тема, дата, канал, статус и текст проблемы.

Вторая может быть заменена или обобщена. Вместо точного адреса достаточно указать регион; вместо возраста — возрастную категорию; вместо полного номера заказа — внутренний код обращения; вместо имени клиента — «Клиент А».

Третья группа не нужна и подлежит удалению. К ней относятся телефоны, электронные адреса, паспортные данные, данные банковских карт, пароли, токены доступа, подписи, полные домашние адреса и фрагменты переписки, не связанные с задачей.

У замены идентификаторов есть важное ограничение. «Клиент А» не делает материал безопасным автоматически. Если в пакете остаются редкая модель товара, точная сумма, дата публичного конфликта и небольшой населённый пункт, человека всё ещё можно узнать по совокупности признаков. Поэтому проверять нужно не только наличие имени, но и возможность восстановить личность по косвенным данным.

Если необходимо связать несколько записей одного клиента, используйте один временный идентификатор внутри текущей задачи. Таблицу соответствия храните отдельно и не передавайте в нейросеть. Если идентификация не нужна, не создавайте такую таблицу вовсе.

До передачи в обработку исходные данные должны пройти несколько преобразований.

Удалить лишнее. Уберите поля и документы, которые не влияют на результат. Например, из задачи о задержке доставки можно удалить телефон, адрес и историю прошлых заказов. Проверка проста: можно ли получить тот же ответ без удалённого поля?

Заменить идентификаторы. Используйте код или условное обозначение вместо имени и номера документа: «Клиент А», «Заказ 1847» — вместо фамилии и полного номера договора. Убедитесь, что в пакете нет отдельной таблицы, связывающей код с конкретным человеком.

Обобщить чувствительные детали. Снизьте точность там, где она не нужна: укажите «Московская область» вместо полного адреса, «сумма до 100 тысяч рублей» вместо точной суммы. При этом проверьте, сохраняется ли аналитическая ценность данных.

Проверить права доступа. Убедитесь, что пользователь и сервис вправе работать с таким типом информации. Перед обработкой кадровых сведений сверьтесь с внутренней политикой и ролью сотрудника. Проверьте, есть ли разрешённый канал и подтверждённый доступ.

Передать минимальный пакет. Отправьте только очищенный материал, цель и ограничения: фрагмент заявки, нужные статусы и правило ответа. Перед отправкой проверьте, не попали ли в запрос скрытые вкладки, вложения и комментарии.

Если задача требует персональных данных в исходном виде, это не повод автоматически их передавать. Сначала нужно подтвердить необходимость, затем проверить полномочия и выбрать разрешённый канал. Для кадровых, медицинских, финансовых и идентификационных сведений планка контроля должна быть выше, чем для обезличенной статистики.

**Сигнал: «У меня есть доступ, значит, я могу отправить»**

Право открыть документ внутри рабочей системы не всегда означает право передать его в любой внешний или внутренний сервис. Для обработки могут действовать отдельные ограничения: по категории данных, цели, сроку хранения, составу пользователей и месту размещения.

Перед началом работы проверьте несколько условий.

Сервис одобрен работодателем для такого типа информации.

Ваша роль позволяет передавать материал именно с этой целью.

Понятно, где хранятся запросы и результаты, кто может их видеть и как долго они сохраняются.

Внутренние правила компании не запрещают передавать коммерческую тайну, персональные данные или материалы по конкретному проекту.

После обработки предусмотрен порядок удаления, архивирования или ограничения доступа.

Если хотя бы один пункт остаётся неизвестным, данные не передают до выяснения. Для рабочей задачи лучше выбрать менее удобный, но разрешённый способ обработки, чем создать неуправляемую копию документа.

**Минимально необходимый объём**

Минимизация — не стремление удалить всё до потери смысла. Это выбор наименьшего объёма данных, который позволяет получить нужный результат.

Решение удобно принимать по дереву условий.

Если задача требует только статистики, передавайте агрегаты, а не строки с персональными данными.

Если нужно сравнить документы, удалите реквизиты, подписи и приложения, не связанные с критериями сравнения.

Если нужно составить ответ конкретному клиенту, оставьте факты обращения и идентификатор заявки, но не передавайте полный профиль клиента.

Если требуется установить личность или проверить право конкретного человека, используйте разрешённую корпоративную систему и подтверждённую роль. Не дублируйте весь профиль в нейросети без отдельной необходимости.

Если документ содержит сведения с разными уровнями доступа, разделите его на части. Не передавайте весь файл только потому, что для работы нужна одна страница.

Если назначение задачи изменилось, соберите контекст заново. Материал, подготовленный для внутренней аналитики, не становится автоматически подходящим для публичного ответа клиенту.

Проверка прав доступа должна охватывать не только пользователя. В процессе есть как минимум три системные роли.

Пользователь определяет цель, отбирает материал и отвечает за первичную очистку.

Администратор настраивает доступы, разрешённые сервисы, группы пользователей, журналы и сроки хранения.

Владелец данных или процесса подтверждает, что правило, документ или набор сведений можно использовать для данной цели.

В небольшой компании один сотрудник может совмещать эти роли, но ответственность всё равно нужно разделить логически. Иначе передача данных превращается в действие без владельца: каждый считает, что проверить должен кто-то другой.

### **Паспорт рабочего контекста**

Чтобы повторяющаяся задача не зависела от памяти одного сотрудника, для неё создают паспорт контекста. Это короткая карточка, которая описывает не только содержание материалов, но и условия их использования.

Минимальный паспорт включает следующие поля.

Название задачи — подготовка черновика ответа по задержке доставки.

Цель использования — дать оператору проверяемый черновик, не принимая решение за руководителя.

Адресат результата — сотрудник поддержки.

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.