

6+

Уволь себя!

Как заставить нейросеть
делать твою работу

Я
СВОБОДНА!



РАБОТАЮ
ЗА ТЕБЯ!



БОЛЬШЕ
ЖИЗНИ



Амелия Белл 

Амелия Белл

**Уволь себя: как заставить
нейросеть делать твою работу**

«Издательские решения»

Белл А.

Уволь себя: как заставить нейросеть делать твою работу /
А. Белл — «Издательские решения»,

А что, если нейросеть перестанет просто советовать — и начнёт работать за вас? Эта книга покажет, как превратить ИИ в цифрового исполнителя: поручать ему документы, таблицы, презентации, поиск, браузер, код и рутину. Без сложного языка и программирования. От первых шагов до ИИ-агентов, которые сами выполняют цепочки задач. Меньше рутины. Больше свободы. Больше жизни.

© Белл А.

© Издательские решения

Содержание

ЧАСТЬ I. ЧАТ, КОТОРЫЙ НЕ РАБОТАЕТ ЗА ВАС	9
Глава 1. Иллюзия готовой работы	9
ЧАСТЬ II. ОТ МОДЕЛИ К ИСПОЛНИТЕЛЮ	14
Глава 4. Семь слоёв цифрового исполнителя	14
Глава 5. Промптинг после эры промптинга	16
Глава 6. Контекст: топливо для реальной работы	18
Глава 7. Память: как не объяснять всё заново	20
Глава 8. Word: от «напиши текст» к готовому документу	22
Глава 9. Excel: не таблица, а модель	24
Глава 10. PowerPoint: делегировать презентацию, а не текст слайдов	26
Глава 11. Почта, календарь и рабочая коммуникация	28
Конец ознакомительного фрагмента.	29

Уволь себя: как заставить нейросеть делать твою работу

Амелия Белл

© Амелия Белл, 2026

ISBN 978-5-0071-3054-7

Создано в интеллектуальной издательской системе Ridero

Вместо предисловия. Нейросеть, которая действительно работает за вас

Когда я впервые увидел, на что способны современные нейросети, всё выглядело почти как фокус.

Можно было написать несколько строк — и через секунды получить письмо, статью, план поездки, рекламный текст или разбор документа. Можно было загрузить таблицу и попросить найти ошибки. Показать фотографию — и спросить, что на ней. Дать длинный текст — и получить короткое объяснение.

Казалось, ещё немного — и компьютер наконец начнёт делать за нас всю скучную работу.

Но довольно быстро обнаружилась странная вещь.

Нейросеть могла прекрасно рассказать, как выполнить работу. Только сама работа почему-то всё ещё оставалась мне.

Допустим, мне нужно подготовить небольшой проект.

Изучить конкурентов. Собрать информацию. Свести цифры в Excel. Написать документ. Сделать презентацию. Подготовить несколько писем. Создать простой сайт. Проверить результат.

Я пишу нейросети: «Сделай всё это за меня».

Через минуту появляется отличный ответ.

Вот анализ рынка. Вот пример таблицы. Вот структура презентации. Вот текст письма. Вот план сайта.

Выглядит впечатляюще.

Но потом я закрываю окно чата.

И вижу, что на моём компьютере почти ничего не изменилось.

Excel-файл никто не создал. Презентации нет. Цифры никто не проверил. Письма не отправлены. В календаре ничего не появилось. Сайт не готов.

Нейросеть рассказала мне, как выполнить работу.

А я хотел, чтобы она выполнила её.

Именно с этого различия начинается эта книга.

Современный искусственный интеллект уже давно умеет гораздо больше, чем просто отвечать на вопросы.

Он может читать файлы, создавать документы и презентации, работать с таблицами, писать программы, искать информацию в интернете, пользоваться внешними сервисами и выполнять действия в браузере.

Он может получать доступ к календарю, почте, облачным дискам, базам данных и другим программам.

А современные ИИ-агенты способны получить большую задачу, разбить её на шаги, выбрать нужный инструмент, выполнить действие, посмотреть, что получилось, и решить, что делать дальше.

Проще говоря, нейросеть постепенно превращается из собеседника в цифрового исполнителя.

И вот это намного интереснее очередного списка «100 лучших промптов».

Промпт — всего лишь способ объяснить машине, чего вы хотите.

Но если у нейросети нет доступа к вашим файлам, программам и рабочим инструментам, даже самый прекрасный промпт не создаст настоящий Excel-файл, не добавит встречу в календарь и не изменит документ.

Чтобы ИИ действительно начал работать за нас, ему нужны своего рода руки.

Таковыми «руками» становятся инструменты, приложения, плагины, API, навыки, MCP, браузер и доступ к программам.

Не пугайтесь этих слов.

Сейчас совершенно не обязательно понимать, что они означают.

Мы дойдём до каждого понятия постепенно.

И я буду объяснять их не языком программистов, а обычным человеческим языком.

Если вы умеете включить компьютер, открыть браузер, найти документ и напечатать текст в Word — этого достаточно, чтобы читать эту книгу.

Начнём мы с самого простого.

С обычного разговора с нейросетью.

Разберёмся, почему она иногда отвечает блестяще, а иногда уверенно говорит ерунду.

Поймём, какие задачи ей можно доверять, а какие необходимо проверять.

Научимся давать ей документы и объяснять, какой результат нам нужен.

После этого начнётся самое интересное.

Мы попробуем сделать так, чтобы результатом работы нейросети был уже не красивый текст в окне чата, а настоящий готовый файл.

Документ Word. Таблица Excel. Презентация PowerPoint. Отчёт. Папка с обработанными файлами.

А затем пойдём дальше.

Научимся подключать к ИИ другие программы.

Разберёмся, что такое плагины и навыки.

Очень простыми словами поймём, что такое API и зачем оно вообще нужно обычному человеку.

Познакомимся с MCP — технологией, которая помогает искусственному интеллекту подключаться к разным инструментам и источникам данных.

Посмотрим, как нейросеть может работать в браузере.

А потом доберёмся до агентов.

Пока достаточно одного простого определения.

ИИ-агент — это нейросеть, которая способна не только ответить вам, но и выполнить последовательность действий для достижения цели.

Например: найти информацию; изучить документы; сравнить данные; заполнить таблицу; создать отчёт; проверить результат; подготовить письмо; и обратиться к вам только там, где действительно требуется ваше решение.

Именно здесь начинается настоящий «Нейромир».

Но есть одна вещь, о которой необходимо договориться заранее.

Я не буду рассказывать вам сказку о волшебном искусственном интеллекте, которому можно бездумно передать всю свою жизнь.

Нейросети ошибаются.

Иногда очень убедительно.

Они могут придумать несуществующий факт. Неправильно прочитать таблицу. Выбрать не тот файл. Нажать не ту кнопку. Сделать разумное действие в неподходящий момент.

И чем больше свободы получает система, тем важнее становится контроль. Поэтому в этой книге мы будем учиться не только давать нейросети больше возможностей.

Мы будем учиться делать это безопасно.

Сначала — посмотреть. Потом — предложить действие. Затем — попробовать на копии. Проверить результат. И только после этого разрешать системе что-либо менять по-настоящему.

Это простое правило убережёт от огромного количества ошибок:

прочитать → предложить → сделать на копии → проверить → выполнить настоящее действие.

Особенно в начале.

В книге будут встречаться названия конкретных нейросетей и моделей.

GPT-6 Astra. Claude Fable 5.1. DeepSeek V4.1 Flash. Gemini 3.8 Flash. И другие.

Эти названия нужны нам как реальные примеры того, что умеют современные системы.

Но запоминать их все не нужно.

Рынок искусственного интеллекта меняется слишком быстро. Сегодня выходит новая модель, через несколько месяцев появляется следующая.

Поэтому сведения о конкретных моделях в этой книге отражают ситуацию на 12 сентября 2026 года.

Названия будут меняться. Кнопки в программах тоже будут меняться. А основные принципы останутся.

Что именно вы хотите получить? Какие данные для этого нужны? Какие инструменты должна использовать система? Что ей разрешено делать? Что ей запрещено? Как проверить результат? И в какой момент решение должен принять человек?

Если вы научитесь отвечать на эти вопросы, переход с одной нейросети на другую уже не будет большой проблемой.

Поэтому не переживайте, если интерфейс программы на вашем экране выглядит немного иначе, чем в книге.

Не ищите глазами каждую кнопку точно на том месте, где она была у меня.

Смотрите глубже.

Что я хочу сделать? Какой файл нужен системе? Какое действие я ей разрешаю? Что должно получиться в конце? Как я это проверю?

Эти вопросы важнее расположения любой кнопки.

И ещё одно.

Не пытайтесь сразу построить огромную систему из десяти агентов, двадцати инструментов и пяти нейросетей.

Возьмите одну маленькую задачу.

Например: привести в порядок документ; обработать небольшую таблицу; разобрать папку с файлами; создать презентацию; подготовить отчёт.

Научитесь передавать машине сначала часть работы.

Потом ещё одну.

Затем соедините эти части.

Так постепенно произойдёт очень интересный переход.

Сначала вы будете спрашивать нейросеть: «Как мне это сделать?»

Потом: «Сделай мне вот эту часть».

Ещё позже: «Выполни всю задачу и покажи результат».

И однажды вы заметите, что главный вопрос изменился.

Вы больше не спрашиваете: «Какой промпт мне написать?»

Вы спрашиваете: «Какую часть этой работы я вообще должен делать сам?»

Вот об этом и написана эта книга.

Как читать эту книгу, если вы боитесь «сломать компьютер»

Для первых опытов вам не нужны API-ключи, программирование и доступ к рабочей почте. Начните с копии обычного файла: документа, таблицы или небольшой папки. Просите ИИ сначала объяснить, что он собирается изменить, затем работать только с копией и в конце показать, что именно поменялось.

Двигайтесь по правилу «прочитать → предложить → сделать на копии → проверить → только потом разрешить реальное действие». Эта последовательность будет повторяться по всей книге. Она важнее названия конкретной модели.

Если в книге встречается функция, которой нет в вашем тарифе или приложении, пропустите конкретную кнопку. Ищите эквивалентную возможность: загрузить файл, создать новый файл, подключить сервис, вызвать инструмент или попросить подтверждение перед действием.

Несколько слов заранее: что я называю агентом

До отдельной главы об агентах нам ещё далеко, но слово будет встречаться раньше. Пока используйте простое определение: агент — это система с ИИ, которая не только отвечает текстом, но и может выбрать следующий шаг, использовать разрешённый инструмент, получить результат и продолжить работу. Например, открыть файл, посчитать данные, создать новый документ или подготовить действие в браузере.

Это не означает, что любой чат уже умеет всё перечисленное. Возможности зависят от конкретного продукта, тарифа, подключённых инструментов и ваших разрешений. Подробно разберём устройство агента в части VII.

ЧАСТЬ I. ЧАТ, КОТОРЫЙ НЕ РАБОТАЕТ ЗА ВАС

Глава 1. Иллюзия готовой работы

Однажды я решил поставить нейросети задачу так, как её обычно ставят хорошему помощнику: «Подготовь мне запуск нового онлайн-сервиса. Изучи рынок, найди конкурентов, сделай финансовую модель, презентацию и план продаж».

Ответ выглядел прекрасно. На экране появились аккуратные заголовки. Затем таблица с предполагаемыми расходами. Потом структура презентации из двенадцати слайдов. В конце — пример письма клиенту.

На чтение ушло меньше десяти минут. На секунду возникло приятное чувство: кажется, половина проекта уже сделана.

А потом я закрыл окно чата и посмотрел на рабочий стол.

Excel-файла не было. Формул не было. Презентации не было. Список конкурентов никто не проверил. Письма никому не отправлены. Сайт не создан. В календаре ничего не появилось. В папке проекта лежали ровно те же файлы, что и раньше.

Нейросеть отлично **описала работу**. Но она не **выполнила работу**.

Это первое различие, которое нам понадобится на протяжении всей книги.

Когда вы просите: «Напиши поздравление», ответ в чате уже может быть готовым результатом. Вы копируете текст и отправляете. Но когда вы просите: «Подготовь месячный отчёт», результатом обычно считается не объяснение, как должен выглядеть отчёт. Нужен файл с правильными данными, формулами, выводами и оформлением. Если вы говорите: «Организуй встречу», хороший совет о том, как искать свободное время, — не результат. Результат — найденный слот и созданное событие в календаре, возможно после вашего подтверждения.

Полезно привыкнуть к очень простому вопросу: **что должно измениться в мире после выполнения задачи?**

Если вы просите рецепт, возможно, достаточно текста. Если вы просите сравнить три договора, нужен вывод, основанный на реальных файлах. Если вы просите «обновить список клиентов», должен измениться файл или запись в CRM. Если просите сделать презентацию — должен существовать файл презентации, который открывается и выглядит нормально.

Именно здесь начинается настоящий разговор об автоматизации.

Ответ и действие

Условно все задачи можно разделить на два типа.

Первый тип — информационный. Вы хотите узнать, понять, придумать или сформулировать. Например: «Объясни простыми словами, что такое ипотека». Здесь текстовый ответ уже является результатом.

Второй тип — исполнительский. Вы хотите, чтобы произошло действие: изменился документ, появилась строка в таблице, была создана презентация, отправлено письмо, запущена программа, заполнена форма на сайте, в календаре появилась встреча.

Современные ИИ-системы способны и на первое, и на второе, но для второго им обычно нужны **инструменты и разрешения**.

Представьте очень умного человека, которого посадили в пустой комнате. У него прекрасная память, он отлично рассуждает и умеет составить план. Но у него нет телефона, компьютера и ключей. Он может объяснить, как заказать такси, но не может заказать его за вас. Как только вы дадите ему телефон и доступ к приложению, ситуация изменится. А если дадите ещё и банковскую карту, появится новая возможность — и новый риск.

С нейросетями происходит то же самое.

Сам по себе интеллект модели — это ещё не действие. Чтобы действовать, система должна иметь доступ к чему-то внешнему: веб-поиску, файлам, API, браузеру, электронной почте, календарю, программе, базе данных или другому инструменту.

Простой тест

Возьмите пять задач, которые вы делали на компьютере за последнюю неделю. Например:

1. ответили на письмо;
2. свели расходы в Excel;
3. нашли информацию о товаре;
4. написали документ;
5. записались к врачу.

Теперь напротив каждой задачи напишите, что является **готовым результатом**.

Для письма это может быть отправленный ответ.

Для работы в Excel — сохранённый файл с верными формулами.

Для поиска товара — сравнительная таблица с актуальными ценами и ссылками.

Для документа — готовый оформленный файл DOCX.

Для записи к врачу — подтверждённое время приёма.

Вы только что сделали первый шаг к автоматизации: перестали описывать работу словами и начали определять её результат.

К этой привычке мы будем возвращаться снова и снова.

Глава 2. Что на самом деле умеет современная модель

Слово «нейросеть» в быту используют почти для всего: от генератора картинок до чат-бота. В этой книге чаще всего речь будет идти о больших языковых моделях — LLM. Не обязательно запоминать английскую аббревиатуру. Важно понять идею.

Такая модель получает информацию и предсказывает, каким должен быть следующий фрагмент ответа. Это звучит скромнее, чем «искусственный интеллект», но результат может быть очень сложным. Модель обучена на огромных массивах текстов, кода, изображений и других данных. Поэтому она умеет замечать структуры: как выглядит деловое письмо, как устроен программный код, как связаны понятия, как преобразовать таблицу в выводы.

Для начинающего пользователя важнее не внутреннее устройство модели, а четыре практических свойства.

Первое: модель умеет работать с языком

Она может объяснять, писать, переписывать, сокращать, переводить, структурировать и сравнивать тексты. Это базовая способность, из которой выросли многие другие.

Если вам трудно начать письмо, модель предложит черновик. Если документ написан слишком сложно — упростит. Если вы получили длинный договор — может помочь разобрать его структуру и обратить внимание на спорные пункты. Последнее не заменяет юриста, но экономит время на первичном чтении.

Второе: модель может работать не только с текстом

Современные системы мультимодальны. Они могут получать изображения, читать документы, анализировать таблицы, в некоторых продуктах — понимать аудио и видео. Это означает, что вы можете не пересказывать вручную содержимое файла, а дать сам файл, если сервис это поддерживает.

Но здесь есть важная ловушка. «Может прочитать» не означает «всегда прочитает без-ошибочно». Плохой скан, сложная таблица, мелкий текст, формула на картинке — всё это повышает вероятность ошибки. Если число критично, проверяйте его в первоисточнике.

Третье: модель может рассуждать о последовательности действий

Если задача сложная, модель способна разбить её на шаги. Например:

«Чтобы подготовить семейный бюджет, сначала соберём доходы и постоянные расходы, затем нерегулярные траты, после этого рассчитаем остаток и резерв».

Это уже похоже на работу консультанта. Но пока это только план.

Четвёртое: модель может выбирать инструменты

И вот здесь начинается особенно интересная часть. Если системе доступны веб-поиск, калькулятор, запуск кода, работа с файлами или внешними приложениями, модель может решить: «Сейчас мне надо не отвечать текстом, а вызвать инструмент».

Например, вы спрашиваете: «Какая сейчас погода и нужно ли брать зонт?» Модель без интернета может только опираться на старые знания или честно сказать, что текущую погоду не знает. Модель с доступом к сервису погоды запросит свежие данные и ответит уже на их основе.

Вы просите: «Посчитай сумму этих расходов». Вместо вычислений «в уме» система может использовать калькулятор или код. Это снижает риск арифметической ошибки.

Вы просите: «Добавь встречу во вторник». Если подключён календарь и разрешено создание событий, модель может вызвать соответствующее действие.

Контекст — рабочий стол модели

Есть ещё одно понятие, которое стоит понять сразу: **контекст**.

Когда мы разговариваем, человеку важно помнить предыдущие фразы. Если я сначала скажу: «У меня красная машина», а потом спрошу: «Какого она цвета?», ответ зависит от того, что собеседник помнит первую фразу.

У модели такую роль рабочей памяти выполняет контекст: текущие сообщения, загруженные файлы, инструкции и другая информация, доступная во время выполнения задачи.

Чем больше полезного контекста, тем лучше система понимает вашу ситуацию. Но «больше» не всегда значит «лучше». Если дать ей десять старых версий одного договора и не сказать, какая актуальная, можно получить очень уверенный вывод на основе неправильного файла.

Почему нейросеть иногда придумывает

Я буду использовать слово «галлюцинация», хотя оно не идеально. Так называют ситуацию, когда модель создаёт правдоподобную, но неверную информацию: придуманную ссылку, несуществующую функцию, ошибочную дату.

Причина проста: модель оптимизирована на создание подходящего ответа, а не на магическое знание истины. Если системе не хватает фактов, она может построить продолжение, которое похоже на правду.

Отсюда главное правило этой главы: **используйте модель как умный механизм обработки информации, но не путайте уверенный стиль с доказательством.**

Если факт важен, дайте системе доступ к актуальному источнику или проверьте источник сами.

Глава 3. Рынок 2026: почему вопрос «какая нейросеть лучшая?» почти бесполезен

К 12 сентября 2026 года рынок снова выглядит иначе, чем всего несколько месяцев назад. OpenAI 3 сентября представила GPT-6 Astra. Anthropic 1 сентября — Claude Fable 5.1. Google 2 сентября выпустила Gemini 3.8 Flash. DeepSeek 10 сентября представила V4.1 Flash. Это четыре релиза за десять дней, и уже один этот факт показывает, почему книга не должна превращаться в рейтинг моделей. [1] [2] [3] [4]

Если я напишу: «модель X — лучшая», такое предложение быстро устареет. Гораздо полезнее научиться выбирать модель под работу.

Представьте, что вы выбираете автомобиль. Фраза «какой автомобиль лучший?» почти бессмысленна без задачи. Для поездки по городу важны одни свойства. Для перевозки мебели — другие. Для бездорожья — третьи. Самая дорогая машина не обязательно лучше выполняет вашу конкретную работу.

С моделями так же.

GPT-6 Astra

OpenAI описывает GPT-6 Astra как модель для сложной сквозной работы, включая рассуждение, программирование, исследования, управление компьютером (computer use) и создание документов. В API на дату написания указано контекстное окно объёмом около 1,05 миллиона токенов и поддержка таких инструментов, как веб-поиск, поиск по файлам и управление компьютером. Доступ после релиза разворачивается постепенно, поэтому наличие модели и отдельных инструментов нужно проверять для своего плана и региона. [1] [5]

Для обычного пользователя важен не сам размер контекстного окна. Важно, что такие модели всё увереннее работают с большими проектами и длинными цепочками действий. Но у этого есть цена: мощная модель может быть дороже и избыточной для простого письма из трёх предложений.

Claude Fable 5.1

Anthropic позиционирует Claude Fable 5.1 как особенно сильную модель для программирования и интеллектуальной работы. Компания отдельно подчёркивает длительные агентные сценарии. Заявленное снижение стоимости типичных нагрузок связано прежде всего с более дешёвым чтением кэша, а не с одинаковым снижением цены всех токенов. [2]

На практике это означает, что Fable интересна там, где задача не заканчивается одним ответом: нужно прочитать много материалов, сделать выводы, работать с кодом или вести длительную работу над проектом.

Gemini 3.8 Flash

Google называет Gemini 3.8 Flash рабочей моделью с акцентом на рассуждение, программную инженерию и агентные задачи. На дату выпуска Google указывала стартовую цену \$0,75 за миллион входных токенов и \$3,75 за миллион выходных токенов; компания отдельно предупреждает, что тариф меняется с 1 января 2027 года. Поэтому цены в этой книге — пример, а не вечный прайс-лист. [4]

Для нас здесь важна идея: иногда лучше взять не самый мощный «флагман», а достаточно сильную и быструю модель, если задача повторяется сотни раз.

DeepSeek V4.1 Flash

DeepSeek V4.1 Flash — хороший пример того, как экономика становится частью архитектуры. Модель использует MoE-подход: общее число параметров велико, но при обработке конкретного запроса активируется лишь часть. DeepSeek заявляет 552 миллиарда параметров и существенно меньшее число активных параметров на входе и выходе, а также нативную работу с изображениями. [3]

Не нужно запоминать цифры. Запомните смысл: при массовых задачах стоимость каждого шага важна. Если агент делает тысячу обращений к модели, разница между дорогим и дешёвым вызовом умножается на тысячу.

Как выбирать без таблицы на сто строк

Я предлагаю пять вопросов.

1. Насколько сложна задача?

Если нужно переписать короткий абзац, флагманская модель часто не нужна. Если нужно провести исследование, проверить много условий и пользоваться инструментами — разумно выбрать более сильную модель.

2. Сколько раз задача повторяется?

Один сложный анализ и десять тысяч классификаций — разные ситуации. Во втором случае стоимость и скорость становятся ключевыми.

3. Нужны ли инструменты?

Модель может быть умной, но если ваш сервис не даёт ей доступ к файлам, браузеру или нужному приложению, большая часть интеллекта останется «в комнате без телефона».

4. Какова цена ошибки?

При выборе названия магазина можно позволить модели больше свободы. Для финансового расчёта нужна проверка, а для действий с деньгами — ещё более строгая.

5. Где находятся ваши данные?

Иногда выбор определяется не моделью, а инфраструктурой компании, требованиями к конфиденциальности или доступными интеграциями.

Поэтому в этой книге мы будем использовать названия моделей как примеры. Но основной вопрос будет другим: **какую систему мы строим вокруг модели?**

ЧАСТЬ II. ОТ МОДЕЛИ К ИСПОЛНИТЕЛЮ

Глава 4. Семь слоёв цифрового исполнителя

Чтобы не утонуть в новых названиях, я предлагаю простую схему. Хороший цифровой исполнитель состоит не из одной нейросети, а из семи слоёв.

Представьте кухню ресторана.

Шеф-повар — это модель. Он умеет думать, планировать и выбирать рецепт. Но без продуктов, ножей, плиты, правил и контроля качества ресторан не заработает.

Слой 1. Модель

Модель — это «мозг» системы. Она понимает задачу, предлагает план, пишет, анализирует и решает, что делать дальше.

Слой 2. Контекст

Это всё, что модель должна знать именно о вашей ситуации: документы, таблицы, переписка, правила, прошлые решения, шаблоны, данные клиента.

Если попросить «сделай отчёт о продажах» и не дать данные о продажах, система либо задаст вопрос, либо начнёт фантазировать. Хороший исполнитель не угадывает то, что можно получить из источника.

Слой 3. Инструкции

Они объясняют, чего вы хотите. Не обязательно писать роман на три страницы. Часто достаточно определить цель, границы и готовый результат.

Например: «Возьми файл продаж за август. Сравни его с июлем. Создай новый Excel-файл с таблицей отклонений и диаграммой. Не меняй исходник. Если найдёшь пустые значения, вынеси их на отдельный лист».

Это уже намного лучше, чем «проанализируй продажи».

Слой 4. Инструменты

Инструмент — это возможность что-то сделать: открыть файл, запустить код, выполнить поиск, создать событие, отправить запрос к системе, изменить таблицу.

Модель может решить, **какой** инструмент использовать, но сам инструмент выполняет действие.

Слой 5. Память и состояние

Если работа длится долго, системе важно помнить, что уже сделано. Например, какие 200 клиентов проверены, на каком этапе остановилась обработка и какие ошибки были исправлены.

Без этого агент похож на сотрудника, который каждое утро просыпается без памяти о вчерашнем дне.

Слой 6. Разрешения

Что системе можно делать?

Читать документы — одно право.

Редактировать — другое.

Удалять — третье.

Отправлять письма — четвёртое.

Совершать покупку — ещё более серьёзное полномочие.

Умная система без ограничений может быть опаснее глупой. Поэтому хорошая архитектура даёт ровно те права, которые нужны для задачи.

Слой 7. Проверка

Как понять, что результат правильный?

Если это файл Excel — открывается ли он, сходятся ли итоги, корректны ли формулы?
Если это презентация — все ли обязательные слайды на месте, нет ли обрезанного текста?
Если это список клиентов — нет ли дублей, заполнены ли нужные поля?
Если это письмо — правильно ли указан адресат и приложен ли нужный файл?

Этот слой часто забывают. Именно поэтому многие красивые демонстрации ИИ плохо переживают встречу с реальной работой.

Соберите свой первый «стек»

Выберите одну простую задачу. Например: «Каждую пятницу делать отчёт о семейных расходах».

Запишите семь строк:

Модель: любая доступная модель.

Контекст: банковская выгрузка и категории расходов.

Инструкция: сравнить неделю с предыдущей.

Инструмент: Excel или другое табличное приложение.

Память: прошлые отчёты.

Разрешение: читать выгрузку и создавать новый файл, но не трогать банковский счёт.

Проверка: сумма категорий должна равняться общей сумме операций.

Вы только что спроектировали маленькую агентную систему, даже если пока ничего не программировали.

Глава 5. Промптинг после эры промптинга

Несколько лет назад вокруг ИИ возникла профессия «промпт-инженер», а интернет заполнили обещания «секретных промптов». Иногда хорошие формулировки действительно дают лучший результат. Но если вы хотите делегировать работу, промпт — лишь одна часть системы.

Я предпочитаю думать не о «магической фразе», а о **техническом задании обычным человеческим языком**.

Хорошее задание отвечает на шесть вопросов.

1. Что нужно получить?

Плохо: «Помоги с бюджетом».

Лучше: «Создай таблицу семейного бюджета на месяц с доходами, постоянными расходами, переменными расходами, накоплениями и итоговым остатком».

2. Из чего это делать?

«Используй приложенную банковскую выгрузку за август и список регулярных платежей».

Если исходных данных нет, никакой красивый промпт их не создаст.

3. Что нельзя делать?

«Не меняй исходный файл. Не придумывай категории, если операция непонятна: пометь её как „проверить“».

Запреты особенно полезны, когда агент умеет действовать.

4. Как выглядит готовый результат?

«Нужен XLSX-файл. На первом листе — сводка, на втором — все операции, на третьем — спорные строки. Итоговая сумма должна совпадать с исходной выгрузкой».

Это называется критерием готовности (definition of done). Термин английский, но идея бытовая. Когда вы говорите мастеру «сделайте ремонт», результат неясен. Когда перечисляете, что именно нужно покрасить и где должны быть розетки, становится понятнее.

5. Что делать, если чего-то не хватает?

Можно написать: «Если без ответа невозможно продолжить, задай один конкретный вопрос. В остальных случаях выбери разумный вариант и укажи допущение».

Это лучше, чем требовать от системы никогда не задавать вопросов или, наоборот, спрашивать разрешение на каждую мелочь.

6. Как проверить себя?

«Перед завершением убедись, что сумма по категориям равна сумме всех операций».

Простая проверка иногда полезнее ещё десяти строк инструкций.

Почему очень длинный промпт может вредить

Новички часто думают: чем длиннее инструкция, тем умнее ответ. На практике длинный промпт может содержать противоречия, лишние требования и устаревшие правила. Модели приходится учитывать всё сразу.

Лучше дать краткое ядро задачи, нужные данные и ясные критерии качества.

Например:

Создай из файла Продажи_август.xlsx краткий отчёт для владельца магазина. Сравни выручку по категориям с июлем, найди пять крупнейших изменений, сделай диаграмму и сохрани новый файл. Исходные файлы не меняй. Любое число в выводах должно быть проверяемо по таблице.

Здесь нет заклинаний. Но есть работа.

Маленькое упражнение

Возьмите фразу, которую вы обычно написали бы ИИ: «Сделай мне презентацию».

Допишите четыре вещи:

- для кого она;
- на основе каких материалов;
- какой файл должен получиться;
- как проверить готовность.

Уже после этого качество результата обычно повышается заметнее, чем после добавления десятка «умных» слов.

Глава 6. Контекст: топливо для реальной работы

Представьте помощника в первый рабочий день. Вы говорите ему: «Подготовь предложение для нашего клиента Иванова».

Если помощник ещё не знаком с клиентом, не знает ваших цен и шаблона, не видел предыдущую переписку, задача почти невыполнима. Одних способностей здесь недостаточно: ему нужен контекст.

С ИИ происходит то же самое.

Что может входить в контекст

Это могут быть:

- файл с требованиями;
- старое коммерческое предложение;
- переписка с клиентом;
- прайс-лист;
- таблица продаж;
- брендбук;
- инструкция компании;
- список запрещённых формулировок;
- предыдущая версия документа;
- база знаний.

Чем точнее данные соответствуют задаче, тем меньше модели приходится угадывать.

Почему нельзя просто «загрузить всё»

Кажется логичным: если контекст полезен, давайте дадим системе всю папку за пять лет. Но это похоже на попытку помочь человеку, высыпав ему на стол все архивные коробки компании.

Проблемы три.

Первая — **устаревшие данные**. В старом прайс-листе цена одна, в новом другая. Если система не знает, какой файл главный, она может взять неправильную.

Вторая — **противоречия**. В одной инструкции написано «ответить за 24 часа», в другой — «за 48 часов».

Третья — шум. Чем больше посторонней информации, тем труднее выделить главное.

Поэтому хороший контекст — не максимальный, а управляемый.

RAG простыми словами

Вы можете встретить термин RAG — Retrieval-Augmented Generation, то есть генерация с поиском по внешним данным.

Представьте библиотекаря. Вы задаёте вопрос. Он не приносит вам всю библиотеку. Сначала находит несколько подходящих книг и страниц, а уже потом отвечает на их основе.

RAG делает примерно то же: система находит подходящие фрагменты в ваших документах и добавляет их в контекст модели.

Это полезно для больших баз знаний, инструкций, договоров и архивов.

Но RAG не гарантирует истину. Если поиск нашёл неправильный документ, модель может аккуратно выстроить вывод на основе неправильного источника. Поэтому важны метаданные: дата, версия, владелец, статус документа.

Контекстный пакет

Для важных проектов полезно собрать маленький «пакет контекста»:

1. короткое описание проекта;
2. цель;
3. актуальные исходные файлы;

4. правила и ограничения;
5. примеры хорошего результата;
6. список терминов;
7. сведения о том, что уже решено.

Такой пакет полезен и человеку. Если новый сотрудник сможет разобраться по нему, скорее всего, и ИИ будет работать лучше.

Глава 7. Память: как не объяснять всё заново

Контекст нужен прямо сейчас. Память нужна, чтобы завтра не начинать работу с нуля. Это различие полезно понимать.

Если вы в чате написали: «Мою кошку зовут Марта», а через пять сообщений спросили: «Как её зовут?», система может помнить это в рамках разговора. Но будет ли она помнить имя через неделю в другом проекте — зависит от продукта и настроек.

В рабочих системах память часто строят не как человеческое воспоминание, а как **сохранённое состояние**.

Что стоит сохранять

Представим, что агент каждую неделю анализирует продажи.

Ему полезно знать:

- где лежат файлы;
- какую версию отчёта считать последней;
- какие категории используются;
- какие ошибки уже были исправлены;
- какой формат предпочитает руководитель;
- какие задачи остались незавершёнными.

Но не всё стоит превращать в «память».

Например, текущий курс валюты лучше каждый раз получать из свежего источника. Текущую цену товара — тоже. Должность сотрудника может измениться. Поэтому динамические факты лучше запрашивать заново, чем надеяться на устаревшее сохранённое значение.

Память может ошибаться

Есть неприятный эффект: сохранённая ошибка становится устойчивой.

Если однажды система записала, что основной клиент — «ООО Альфа», хотя это уже не так, и потом месяцами использует эту информацию, ошибка начинает влиять на новые документы.

Поэтому рабочая память должна быть управляемой: должно быть видно, откуда пришёл факт, когда он обновлялся и каким источником его при необходимости заменить.

Память и конфиденциальность

Чем больше система помнит о вас и компании, тем выше ценность этой памяти — и тем выше требования к защите.

Не храните секреты просто потому, что «так удобнее». Пароли, ключи API, данные банковских карт и другие чувствительные сведения должны жить в предназначенных для этого защищённых системах, а агенту следует давать только необходимый доступ.

Журнал важнее памяти

Для серьёзной автоматизации я бы выделил отдельную вещь — **журнал действий**.

Память говорит: «что мы знаем».

Журнал говорит: «что мы сделали».

Например:

- 09:02 — получен файл Продажи_август. xlsx.
- 09:03 — обнаружено 812 строк.
- 09:04 — 7 строк без категории вынесены на проверку.
- 09:06 — создан новый файл отчёта.
- 09:07 — итоговая сумма сверена с источником.

Когда всё хорошо, журнал кажется скучным. Когда что-то пошло не так, он становится бесценным.

Теперь у нашего цифрового исполнителя есть мозг, контекст и память. В следующей части не будем сразу погружаться в API и протоколы: сначала посмотрим на то, что новичку знакомо лучше всего — Word, Excel, PowerPoint, почту и обычные файлы. Так легче увидеть практическую пользу, а уже затем понять технический механизм.

ЧАСТЬ III. ПЕРВЫЕ РЕАЛЬНЫЕ РЕЗУЛЬТАТЫ: WORD, EXCEL, POWERPOINT И ФАЙЛЫ

Глава 8. Word: от «напиши текст» к готовому документу

Word кажется простым приложением, потому что большинство людей используют его как печатную машинку: открыли документ, набрали текст, сохранили файл. Но настоящий рабочий документ — это не только слова.

У него есть структура, заголовки, таблицы, стили, нумерация, ссылки, иногда колоннотулы, комментарии и версия для согласования.

Поэтому задача «напиши отчёт» и задача «подготовь готовый Word-документ» сильно отличаются.

Перед началом проверьте простую вещь: умеет ли выбранный сервис действительно создавать или редактировать файл DOCX. Если он может только написать текст в чате, результатом будет текст, а не готовый Word-файл. Тогда либо перенесите текст в Word сами, либо используйте инструмент или приложение, которому разрешена работа с документами.

Полный цикл

Представим, что нам нужен отчёт о проекте.

Хороший агентный процесс выглядит так:

1. Прочитать исходные материалы.
2. Выделить факты и решения.
3. Построить структуру.
4. Написать черновик.
5. Проверить, что выводы опираются на данные.
6. Применить шаблон компании.
7. Добавить таблицы.
8. Проверить заголовки и нумерацию.
9. Сохранить DOCX.
10. Открыть результат и убедиться, что ничего не «съехало».

Последний шаг важнее, чем кажется. Документ может быть технически создан, но таблица вышла за поля страницы или заголовков остался внизу страницы без следующего абзаца.

Microsoft 365 и агентный режим

В апреле 2026 года Microsoft объявила, что агентные возможности Copilot в Word, Excel и PowerPoint стали общедоступны. Компания описывает эти возможности как способность выполнять многошаговые действия прямо внутри документа, таблицы или презентации. [13]

Это хороший пример общего движения рынка: ИИ больше не ограничивается боковой панелью с советами. Он получает возможность менять сам объект работы.

Простой пример

Допустим, вы управляете маленькой школой и хотите сделать инструкцию для новых учеников.

Вместо команды «Напиши инструкцию для учеников» дайте системе следующие исходные материалы:

- старую инструкцию;
- расписание;
- список контактов;
- логотип;
- требования к оформлению.

Поставьте задачу:

«Создай двухстраничный Word-документ для новых учеников. Используй простой язык. Сделай разделы „Первый день“, „Расписание“, „Что взять с собой“, „К кому обратиться“. При-

меняй заголовки Word, а не просто жирный текст. В конце проверь, что телефоны и часы работы взяты из приложенных материалов, а не придуманы».

Так мы делегируем документ, а не отдельный абзац.

Где человек всё ещё полезен

Есть вещи, которые трудно формализовать: дипломатический тон, чувствительные политические темы, скрытая история отношений между людьми.

В таких ситуациях ИИ может подготовить основу, но финальный выбор формулировок остаётся за человеком.

Автоматизация не требует убрать человека отовсюду. Она требует освободить его от механических шагов, где его внимание почти ничего не добавляет.

Глава 9. Excel: не таблица, а модель

Excel — одна из областей, где ИИ может принести огромную пользу и одновременно создать очень убедительную ошибку.

Причина проста: таблица с красивыми числами выглядит серьёзно. Но если формула неверна, красота только маскирует проблему.

Поэтому к Excel мы подойдём медленно.

Что ИИ может делать с таблицей

Современный агент может помочь, но для первых запусков лучше работать с копией книги Excel и сохранять исходный файл без изменений:

- импортировать данные;
- привести даты к одному формату;
- очистить лишние пробелы в рабочей копии;
- найти и пометить дубликаты;
- добавить формулы;
- построить сводную таблицу;
- сделать диаграмму;
- предложить возможные причины изменения показателя;
- сравнить сценарии;
- найти подозрительные значения.

В агентном режиме Excel Microsoft прямо описывает возможность выполнять анализ, объяснять его результаты и вносить изменения в книгу — от формул до таблиц и визуализаций.

[13]

Наш сквозной пример: маленький онлайн-сервис

Предположим, мы запускаем сервис с подпиской за 15 евро в месяц.

Нам нужно оценить:

- сколько клиентов придёт;
- сколько заплатят;
- сколько клиентов уйдёт;
- сколько стоит реклама;
- сколько стоят серверы и поддержка;
- когда проект выйдет в плюс.

Новичок часто просит: «Сделай финансовую модель».

Это слишком расплывчато.

Лучше перечислить допущения:

Цена: 15 евро.

Старт: 100 клиентов.

Новые клиенты: 10% от числа активных клиентов в начале месяца.

Отток: 4% от той же активной базы в начале месяца.

Реклама: 2000 евро в месяц.

Прочие постоянные расходы: 1500 евро.

Тогда логика расчёта числа клиентов прозрачна: клиенты следующего месяца = клиенты в начале месяца + новые клиенты — ушедшие клиенты. В таком учебном прогнозе дробное число клиентов можно трактовать как ожидаемое значение. Если же вы говорите «рост 10%» и уже имеете в виду чистый рост после оттока, отдельно вычитать ещё 4% нельзя — получится двойной учёт.

Также заранее задайте момент, к которому относится выручка. Если подписка списывается в начале месяца, одна формула будет уместна; если люди приходят и уходят в течение

месяца, для прогноза может понадобиться среднее число активных клиентов. Важна не сложность формулы, а явное правило времени.

Затем попросите создать лист «Допущения», где все числа, которые можно менять, вынесены отдельно. Это важный принцип хорошей модели: не прятать исходные предположения внутри формул.

Формула должна быть проверяема

Допустим, агент рассчитал выручку месяца как:

клиенты × цена.

Это понятно.

Но если он создаёт сложную формулу из двадцати вложенных функций, а вы не понимаете, что она делает, риск растёт.

Для важных таблиц я рекомендую просить три вещи:

1. Отдельно перечислить допущения.
2. Подписать ключевые формулы человеческим языком.
3. Сделать контрольные проверки.

Например:

«Сумма расходов по категориям должна равняться общим расходам».

Сводные таблицы и диаграммы

ИИ особенно полезен для людей, которые никогда глубоко не изучали Excel.

Вы можете попросить:

«Покажи продажи по месяцам и категориям. Сделай сводную таблицу и диаграмму.

Отдельно выдели три категории с самым сильным падением».

Но не забывайте: вывод должен быть связан с реальными данными.

Если продажи упали на 20%, это факт.

Почему они упали — уже гипотеза. Возможно, сезонность. Возможно, цена. Возможно, закончился товар. Таблица сама не доказывает причину.

Проверка Excel-файла

Перед тем как доверять созданной книге, задайте пять вопросов:

- открывается ли файл без ошибок;
- есть ли ожидаемые листы;
- верны ли ссылки формул;
- сходятся ли контрольные суммы;
- не подменены ли реальные данные придуманными.

Если речь идёт о деньгах, я бы добавил независимую проверку ключевых итогов калькулятором, кодом или альтернативной формулой.

Что меняется для новичка

Раньше человек мог годами не пользоваться сводными таблицами, потому что не знал, где находится нужная кнопка.

Теперь путь другой. Вы описываете результат естественным языком, а агент выполняет механическую часть. Ваша новая задача — не помнить каждый пункт меню, а понимать, **какой анализ имеет смысл и как проверить ответ**.

Глава 10. PowerPoint: делегировать презентацию, а не текст слайдов

Есть простой способ получить плохую презентацию: попросить ИИ «сделать десять слайдов о нашей компании».

Система честно создаст десять слайдов. Но число слайдов — не цель презентации.

Перед хорошей презентацией нужно ответить на три вопроса:

- кто будет смотреть;
- что этот человек должен понять;
- какое решение должно стать проще после просмотра.

Презентация — это последовательность аргументов

Допустим, мы хотим убедить партнёра вложиться в наш сервис.

Плохая структура:

1. О компании.
2. Наша миссия.
3. Наш продукт.
4. Рынок.
5. Команда.

Это набор тем.

Более сильная структура может быть такой:

1. У клиента есть дорогая проблема.
2. Сейчас её решают плохо.
3. Наш продукт сокращает конкретные издержки.
4. Вот данные пилота.
5. Вот размер доступного рынка.
6. Вот экономика.
7. Вот риски.
8. Вот что нам нужно для следующего этапа.

Это уже аргумент.

Что агент может сделать

Если дать ему исходный отчёт и Excel-модель, он может:

- выбрать ключевые выводы;
- предложить логику презентации (storyline);
- создать слайды;
- перенести диаграммы;
- применить шаблон;
- добавить заметки выступающего;
- проверить, что числа совпадают между презентацией и таблицей.

Microsoft в 2026 году описывает агентное создание и редактирование прямо внутри Word, Excel и PowerPoint; в разных планах и интерфейсах название режима может отличаться. В документации Office Agent также описан многоагентный подход. Для читателя важнее функция — создать и доработать реальный файл, — чем точное название кнопки. [14]

Визуальный контроль обязателен

Презентация — визуальный объект. Недостаточно проверить текст программно.

Откройте файл и пролистайте все слайды.

Смотрите:

- не выходит ли текст за границы слайда;
- читаются ли подписи;

- не слишком ли много слов;
- соответствуют ли диаграммы данным;
- одинаково ли оформлены похожие элементы.

Если агент умеет сам создавать изображения слайдов и просматривать их, поручите ему и эту проверку. Если нет — сделайте её вручную.

Простая команда для хорошего результата

Вместо:

«Сделай презентацию»

попробуйте:

«Изучи отчёт и финансовую модель. Сначала сформулируй четыре вывода, которые важны для владельца бизнеса. Затем построй презентацию вокруг этих выводов. Не добавляй число, которого нет в источниках. На каждом слайде должна быть одна основная мысль. Создай РРТХ и проверь визуально каждый слайд».

В этой команде главное слово — не РРТХ. Главное — **выводы, источники и проверка.**

Глава 11. Почта, календарь и рабочая коммуникация

Электронная почта — идеальный кандидат на автоматизацию, потому что в ней много повторяющейся работы. Но это и опасная область: одно ошибочно отправленное письмо может испортить отношения.

Поэтому здесь особенно полезно разделять **подготовку** и **отправку**.

Что можно отдать ИИ почти сразу

- найти письма по теме;
- собрать длинную цепочку в краткое резюме;
- выписать вопросы, на которые никто не ответил;
- найти обещанные сроки;
- подготовить черновик;
- перевести письмо;
- изменить тон.

Это действия с низкой ценой ошибки, если вы просматриваете результат.

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.