

САВЕЛИЙ ОРЛОВ

ИИ-АГЕНТЫ БЕЗ КОДА

Как собрать помощника
под свою задачу

20

ГОТОВЫХ
СЦЕНАРИЕВ



- ☒ Определить триггер
- ☒ Составить шаги
- ☒ Настроить действие
- ☒ Протестировать
- ☒ Запустить

Савелий Орлов

**ИИ-агенты без кода. Как собрать
помощника под свою задачу**

«Автор»

2026

Орлов С.

ИИ-агенты без кода. Как собрать помощника под свою задачу /
С. Орлов — «Автор», 2026

Вы уже в пятый раз копируете данные из формы в чат, результат — в таблицу, потом открываете почту и запускаете следующий запрос. Через двадцать минут всё готово. А в среду — повторить. И в пятницу тоже. Проблема уже не в промптах. Проблема в том, что между сервисами до сих пор ходите вы. Эта книга — практическое введение в LLM, промпт-инжиниринг, автономные агенты и автоматизацию без программирования. Вы разберётесь, какие задачи стоит отдавать агентам, как раскладывать процессы на шаги, где они ломаются и когда автоматизация просто не окупается. Затем соберёте своего первого агента: подключения, ключи доступа, проверки и запуск. А в финале получите 20 готовых сценариев — от обработки заявок и документов до контента, сводок и поиска по собственным данным. Без обещаний «заменить отдел за вечер». С честным разбором ограничений и границы между no-code и программированием. Актуально на 2026 год. Главное обещание книги — один работающий агент под вашу реальную задачу.

© Орлов С., 2026

© Автор, 2026

Содержание

Введение. Вы делаете это в пятый раз	7
Что такое агент, в одном предложении	8
Обещание	9
Важное предупреждение про эту книгу	10
Кому эта книга не подойдёт	11
Чего в книге нет	12
Как читать	13
Часть I. Как думать про агентов	14
Глава 1. Что такое агент на самом деле	15
Ситуация	15
Определение без терминов	16
Чем это отличается от чата	17
Три составляющие	18
Триггер: что запускает	18
Шаги: что происходит	18
Действие: что получается	18
Разбор трёх примеров	19
Пример 1. Заявки с сайта	19
Пример 2. Утренняя сводка	19
Пример 3. Разбор входящих документов	19
Почему «умный собеседник» — вредная метафора	21
Чего агент не умеет	22
Где эта глава не поможет	23
Итог главы	24
Глава 2. Какие задачи годятся	25
Ситуация	25
Четыре критерия	26
1. Задача повторяется	26
2. Шаги одинаковы	26
3. Результат проверяем	26
4. Ошибка обратима	26
Что реально работает	27
Что не работает	28
Главная ошибка: слишком большая задача	29
Как выбрать первую	30
Таблица для отбора	31
Где эта глава не поможет	32
Итог главы	33
Глава 3. Разложить задачу на шаги	34
Ситуация	34
Почему это главное	35
Правило одного глагола	36
Вход и выход каждого шага	37
Развилки	38
Главная ошибка: шаг «модель разберётся»	39
Как раскладывать, если не получается	40

Способ 1. Проговорить вслух	40
Способ 2. Интервью с самим собой	40
Способ 3. Пойти от результата	40
Разбор: одна задача, две раскладки	41
Раскладка первая, плохая	41
Раскладка вторая, рабочая	41
Второй пример: там, где считают	43
Чек-лист готовой раскладки	44
Где это не сработает	45
Конец ознакомительного фрагмента.	46

Савелий Орлов
ИИ-агенты без кода. Как собрать
помощника под свою задачу

ИИ-агенты без кода

Как собрать помощника под свою задачу

Савелий Орлов

Серия «Нейросети на практике»

Введение. Вы делаете это в пятый раз

Понедельник, девять утра. Вы открываете чат и вставляете в него запрос, который сохранён у вас в записках.

Он хороший, этот запрос. Вы его отлаживали, в нём есть роль, контекст, формат и ограничения. Он работает.

Дальше вы копируете данные из формы. Вставляете. Получаете результат. Копируете его в таблицу. Открываете почту, берёте оттуда три письма, вставляете следующий запрос. Правите, что получилось. Отправляете.

Двадцать минут.

В среду вы делаете это снова, в пятницу снова — заявки копятся не только к понедельнику. И в следующий понедельник всё заново.

Проблема уже не в том, что вы не умеете формулировать задачу — умеете, у вас целая библиотека таких запросов. Проблема в том, что между запросами есть вы: копируете, вставляете, переносите из окна в окно и не забываете запустить в нужный день.

Всё это можно записать один раз и больше не делать.

Что такое агент, в одном предложении

Без терминов, без английских слов, без разговоров об автономных системах:

Агент — это записанная последовательность шагов, в которой один-два шага делает модель.

Взять данные оттуда. Разобрать их. Положить сюда. Сообщить вам. Записанное один раз и выполняющееся само.

Механических шагов — взять, переложить, отправить — окажется в несколько раз больше, чем умных. И это хорошая новость: значит, учиться нужно не работе с искусственным интеллектом, а тому, как разложить свою работу на шаги.

Такое разложение — обычный человеческий навык, и он полезен далеко за пределами этой книги.

Обещание

Одно, узкое и проверяемое:

Один работающий агент под вашу задачу, собранный вами, без единой строки кода.

Не «армия помощников». Не «бизнес на автопилоте». Не «ИИ-сотрудник, который работает круглосуточно». Один агент, который делает то, что вы делаете руками в пятый раз.

Когда он заработает, вы соберёте второго за вечер — потому что весь трудный путь пройден на первом.

Важное предупреждение про эту книгу

Книга состоит из трёх частей, и они устаревают с разной скоростью. Скажу об этом сразу, чтобы вы понимали, чему доверять через год.

Часть первая — как думать про агентов. Что это такое, какие задачи годятся, как разложить задачу на шаги, где всё ломается, сколько стоит. Здесь нет ни одного названия сервиса и ни одного описания интерфейса. Эта часть не устареет: она про способ мышления, а он от версий не зависит.

Часть вторая — как собрать. Пошаговая сборка на конкретном инструменте. Вот эта часть привязана ко времени. Конструкторы меняют интерфейсы, тарифы и названия кнопок каждые несколько месяцев.

Я старался описывать логику действия, а не расположение кнопок, — так текст живёт дольше. Но если вы читаете это через год-полтора и видите на экране не то, что описано, — верьте экрану, а не книге. Логика останется той же, названия поменяются.

Часть третья — каталог сценариев. Двадцать готовых агентов под конкретные задачи. Устаревает медленно: задачи людей меняются гораздо реже, чем интерфейсы.

Как проверить актуальность, если сомневаетесь: откройте раздел помощи того сервиса, который выберете, и сверьте названия основных действий. Пять минут.

Кому эта книга не подойдёт

Тем, кто ещё не пользуется нейросетями регулярно. Агент автоматизирует то, что вы уже умеете делать руками. Если вы пока не составляете запросы и не знаете, чего от модели ждать, начинать с агентов рано — сначала книга «Нейросети для работы», там разобрана механика, на которую здесь всё опирается.

Тем, кто ищет готовое решение под ключ. Здесь про то, как собрать самому. Если задача сложная и денег больше, чем времени, дешевле заказать.

Программистам. Вам будет скучно: всё, что здесь описано, вы сделаете кодом быстрее и гибче.

Чего в книге нет

Обещаний автономности. Каждый раз, когда речь зайдёт о том, что агент работает без вас, рядом будет оговорка про контроль. Агент без присмотра ломается молча — это его нормальное поведение.

Историй про сокращённые отделы. Не знаю таких и не буду выдумывать.

Прогнозов. Что будет с агентами через пять лет — не знаю.

Обещаний, что это просто. Первый агент займёт у вас вечер или два, и на середине вы один раз захотите всё бросить. Это нормальный ход событий, а не признак того, что у вас не получается.

Как читать

Часть первую — целиком и подряд, до того как что-либо открывать. Пять глав, около двух часов. Половина неудачных агентов проваливается ещё до сборки, и все причины разобраны там.

Особенно третья глава: раскладка задачи на шаги — навык, который решает всё остальное.

Часть вторую — с ноутбуком. Читать её без экрана бессмысленно, а собирать без части первой — рано.

Часть третью — по своей задаче. Найдите ближайший сценарий и стройте от него.

Если вы человек, который учится с конца, — начните с третьей части, найдите похожий сценарий и возвращайтесь в первую за объяснением, почему он устроен именно так.

Часть I. Как думать про агентов

Пять глав до того, как вы что-либо откроете. Здесь нет ни одного названия сервиса и ни одного описания интерфейса — эта часть про способ мышления, и она не устареет.

Половина неудачных агентов проваливается ещё до сборки, и все причины разобраны здесь.

Глава 1. Что такое агент на самом деле

Ситуация

Вернёмся к тому понедельнику из введения.

Вы уже умеете формулировать запросы — они у вас сохранены, вы просто вставляете их и меняете данные. Проблема не в том, что вы не знаете, что просить. Проблема в том, что вставлять и копировать приходится вам.

Вот здесь и начинается эта книга. Всё, что вы делаете руками между запросами, — копирование, вставка, перенос из одного окна в другое, запуск в нужное время, — можно записать один раз и больше не делать.

Записанная последовательность этих действий и называется агентом.

Определение без терминов

Начну с формулировки, на которой стоит вся книга:

Агент — это записанная последовательность шагов, в которой один-два шага делает модель.

Не «умный помощник». Не «искусственный сотрудник». Последовательность шагов, записанная один раз и выполняющаяся сама.

Разберу по частям, потому что каждое слово здесь важно.

«Последовательность шагов». Взять данные оттуда, сделать с ними то, положить сюда, сообщить об этом туда — обычная цепочка действий, которую вы сейчас выполняете вручную.

«Записанная». Один раз описанная так, чтобы повторялась без вас. Не «объяснённая» — записанная, буквально, по шагам.

«В которой один-два шага делает модель». Вот единственное отличие от обычной автоматизации, которая существует лет тридцать. Один из шагов — тот, где нужно понять текст, составить ответ, разложить по смыслу, — выполняет нейросеть.

Остальные шаги никакого отношения к искусственному интеллекту не имеют. Взять письмо из почты, записать строку в таблицу, отправить уведомление — это простые механические действия.

Практическое следствие, ради которого всё это сформулировано. Механических шагов в агенте обычно в несколько раз больше, чем умных. Значит, и учиться нужно в первую очередь не работе с моделью, а тому, как разложить задачу на шаги.

Этому посвящена третья глава, и она главная в книге.

Чем это отличается от чата

Вы уже пользуетесь нейросетью. Разница между тем, что вы делаете сейчас, и агентом — в трёх вещах.

Первое: он запускается без вас.

В чате запуск — это вы. Открыли, вставили, нажали. Агент запускается сам: по времени, по событию, по появлению нового письма или новой строки в таблице.

Это и есть главное отличие. Всё остальное — следствия.

Второе: он берёт данные сам.

В чате данные приносите вы: копируете и вставляете. Агент берёт их из источника напрямую — из почты, из таблицы, из формы на сайте, из сообщения в мессенджере.

Третье: он делает несколько шагов подряд.

В чате результат — это текст в окне. Дальше вы решаете, что с ним делать. Агент доводит цепочку до конца: получил, обработал, положил результат в нужное место, сообщил вам.

Всё вместе: вы перестаёте быть переносчиком данных между окнами. Роль, которую вы играете сейчас, — курьер между формой, чатом и таблицей — исчезает.

Три составляющие

Любой агент, независимо от сложности и от инструмента, состоит из трёх частей. Это самая полезная схема первой части книги — по ней вы будете разбирать и свои задачи, и чужие примеры.

Триггер: что запускает

Событие, после которого агент начинает работать.

Четыре типовых варианта:

По расписанию. Каждый день в девять утра. Каждый понедельник. Первого числа каждого месяца.

По появлению нового. Пришло письмо. Заполнили форму. Появилась строка в таблице. Загрузили файл в папку.

По команде. Вы отправили сообщение боту или нажали кнопку.

По изменению. Значение в таблице превысило порог. Статус заказа поменялся.

Триггер выбирается первым, и от него зависит вся конструкция. Один и тот же агент с триггером «по расписанию» и «по появлению письма» — это два разных агента с разной логикой.

Шаги: что происходит

Последовательность действий между запуском и результатом.

Шаги бывают трёх видов:

Механические. Взять, положить, скопировать, отправить, посчитать по формуле. Никакого интеллекта не требуют, работают всегда одинаково.

Умные. Разобрать текст, составить ответ, разложить по категориям, извлечь смысл, сформулировать. Здесь работает модель — и здесь же появляется всё, что вы знаете о ненадёжности её ответов.

Проверочные. Убедиться, что результат похож на правду, и решить, что делать дальше. Ими люди чаще всего пренебрегают, и напрасно — четвёртая глава об этом.

Механических обычно в несколько раз больше. Если у вас получается наоборот, задача, скорее всего, разложена плохо.

Действие: что получается

То, ради чего всё затевалось. Результат должен куда-то попасть.

Куда обычно: строка в таблице, письмо адресату, сообщение вам, файл в папке, запись в системе, уведомление в мессенджер.

Важно: результат «текст на экране» для агента не годится. Если никто не смотрит на экран в момент срабатывания, результат исчез. Агент должен положить результат туда, где вы его найдёте.

Разбор трёх примеров

Три бытовые задачи, разобранные по одной схеме. Смотрите не на содержание, а на то, как они раскладываются.

Пример 1. Заявки с сайта

Задача. Люди заполняют форму на сайте. Вы вручную переносите заявки в таблицу и отвечаете каждому.

Триггер: заполнена форма.

Шаги:

Взять данные из формы (*механический*)

Определить тип заявки: срочная, обычная, не по адресу (*умный*)

Записать строку в таблицу с типом (*механический*)

Составить черновик ответа под тип заявки (*умный*)

Отправить черновик мне в мессенджер (*механический*)

Действие: строка в таблице и готовый черновик у меня в телефоне.

Обратите внимание на пятый шаг. Агент не отправляет ответ клиенту сам — он присылает черновик мне. Это осознанный выбор: цена ошибки в письме клиенту высокая, а проверка занимает десять секунд. Про то, когда нужен человек в цепочке, — девятая глава.

Пример 2. Утренняя сводка

Задача. Каждое утро вы просматриваете почту, календарь и таблицу с задачами, чтобы понять, что сегодня.

Триггер: каждый день в 8:30.

Шаги:

Взять непрочитанные письма за сутки (*механический*)

Взять события из календаря на сегодня (*механический*)

Взять задачи со сроком сегодня и завтра (*механический*)

Собрать из этого короткую сводку: что срочно, что можно отложить, о чём не забыть (*умный*)

Отправить сводку в мессенджер (*механический*)

Действие: одно сообщение в телефоне в 8:30.

Здесь один умный шаг из пяти. Ценность агента не в интеллекте, а в том, что он сходил в три места вместо вас.

Пример 3. Разбор входящих документов

Задача. Вам присылают счета в почту, вы вручную выписываете из них суммы и даты в таблицу.

Триггер: пришло письмо с вложением определённого типа.

Шаги:

Взять вложение (*механический*)

Извлечь текст из документа (*механический*)

Найти в тексте номер, дату, сумму, поставщика (*умный*)

Проверить, что сумма — число, а дата похожа на дату (*проверочный*)

Записать строку в таблицу (*механический*)

Если проверка не прошла — прислать мне письмо целиком (*механический*)

Действие: заполненная строка или уведомление о том, что разобрать не удалось.

Четвёртый и шестой шаги — то, что отличает работающий агент от красивого.

Без них агент, встретив нестандартный документ, запишет в таблицу правдоподобную чушь, и вы узнаете об этом через квартал.

Почему «умный собеседник» — вредная метафора

Самое распространённое представление об агенте: это как чат, только он сам всё делает. Из этого представления следуют три ошибки, каждая из которых стоит недель.

Ошибка первая: ожидание понимания. Собеседнику можно сказать «разберись с почтой», и он сообразит. Агенту — нельзя. Он делает ровно то, что записано, и ничего сверх. Шаг «модель разберётся» — самая частая причина, по которой первый агент не работает.

Ошибка вторая: ожидание гибкости. Собеседник, столкнувшись с необычной ситуацией, изменит поведение. Агент выполнит записанный шаг и на необычных данных — с результатом, который никто не предсказывал. Все нестандартные случаи должны быть предусмотрены заранее, иначе они не предусмотрены вообще.

Ошибка третья: ожидание ответственности. С собеседника можно спросить. С агента — нет. За то, что он сделал, отвечаете вы, включая случаи, когда он сделал это в три часа ночи, а вы узнали через неделю.

Полезная метафора вместо вредной: агент — не сотрудник, а конвейер. Конвейер работает быстро и одинаково, не устаёт и не отвлекается. И он не заметит, что деталь на входе была бракованной, если его не научили это проверять.

Чего агент не умеет

Короткий список, который стоит держать в голове всю книгу.

Понимать намерение. Он не знает, зачем вы это делаете, и не догадается, что в этот раз нужно иначе.

Менять план. Записанная последовательность не перестраивается по ситуации. Если ситуация изменилась, перестраиваете вы.

Оценивать важность. Для агента все входящие одинаковы, пока вы не научили его различать — и научить можно только по формальным признакам, а не по смыслу «это важный клиент».

Отвечать за результат. Ответственность не делегируется вместе с задачей.

Работать с тем, чего не было при сборке. Изменился формат входящих данных — агент сломается или, хуже, продолжит работать неправильно. Об этом четвёртая глава.

Где эта глава не поможет

Если вы ещё не пользуетесь нейросетями регулярно. Агент — это автоматизация того, что вы уже умеете делать руками. Если вы не умеете формулировать запрос, автоматизировать нечего. Начните с книги «Нейросети для работы».

Если задача не повторяется. Разовую задачу быстрее сделать руками, чем описать. Критерии — во второй главе.

Если вам нужна сложная логика. Есть предел того, что собирается без кода. Он высокий, но он есть, и в восьмой главе я скажу прямо, где он проходит.

Итог главы

Агент — записанная последовательность шагов, в которой один-два шага делает модель.

Отличие от чата: запускается сам, берёт данные сам, доводит цепочку до конца.

Вы перестаёте быть переносчиком данных между окнами.

Три составляющие: триггер, шаги, действие. По этой схеме разбирается любой агент.

Триггер выбирается первым, от него зависит вся конструкция.

Шаги бывают механические, умные и проверочные. Механических обычно в несколько раз больше.

Результат должен куда-то попасть. «Текст на экране» для агента не результат.

Проверочные шаги отличают работающий агент от красивого.

«Умный собеседник» — вредная метафора: агент не поймёт намерение, не проявит гибкость и не понесёт ответственность.

Полезная метафора — конвейер: быстро, одинаково, без усталости и без внимания к браку на входе.

Что сделать сегодня. Возьмите задачу, которую вы делаете каждую неделю через чат, и разложите её на бумаге по трём составляющим: что запускает, какие шаги, куда попадает результат. Не пытайтесь ничего собирать — просто разложите. Это ровно тот навык, который решает всё остальное, и вторая глава поможет понять, годится ли выбранная задача вообще.

Следующая глава — про то, какие задачи стоит брать, а какие не стоит. Половина неудачных агентов проваливается именно здесь, ещё до первого шага сборки.

Глава 2. Какие задачи годятся

Ситуация

Вы прочитали первую главу, поняли, что такое агент, и вам сразу пришла в голову задача.

Обычно это самая раздражающая работа из всех, что у вас есть. То, от чего вы устали. То, что вы отложили бы навсегда, если бы могли.

С высокой вероятностью это неподходящая задача.

Раздражает обычно то, что требует внимания и решений, — а именно это агенту не отдаётся. Подходящая задача чаще всего скучная, регулярная и не особенно вас беспокоящая. Вы её просто делаете, не задумываясь, сколько времени она съедает.

Эта глава — про то, как выбрать первую задачу. Половина неудачных агентов проваливается здесь, ещё до раскладки.

Четыре критерия

Задача годится, если выполняются все четыре условия. Не три, а четыре.

1. Задача повторяется

Не «бывает время от времени», а происходит с понятной регулярностью: каждый день, каждую неделю, каждый раз при определённом событии.

Почему это первое условие. Сборка первого агента займёт вечер-два. Эти часы возвращаются только за счёт повторов. Задача раз в квартал не окупится никогда — вы забудете, как агент устроен, раньше, чем он отработает третий раз.

Ориентир по частоте. Ежедневно или чаще — отлично. Еженедельно — хорошо. Раз в месяц — сомнительно. Реже — не берите.

Важнее частоты — количество событий. Задача, которая случается по двадцать раз в день, окупится, даже если каждое занимает минуту.

2. Шаги одинаковы

Каждый раз вы делаете примерно одно и то же в одном и том же порядке.

Проверка: можете ли вы описать эту работу человеку, которого нанимаете на неё завтра? Если да — шаги одинаковы. Если вы говорите «ну, тут по ситуации» — нет.

Частая ловушка. Кажется, что шаги одинаковы, а на деле вы каждый раз принимаете десяток мелких решений на автомате, не замечая этого. Раскладка из третьей главы это вскроет.

3. Результат проверяем

Вы должны уметь быстро понять, правильно агент отработал или нет.

Проходит: записанная строка в таблице — видно, что там. Черновик письма — прочитали и оценили. Сводка — просмотрели.

Не проходит: результат, для проверки которого нужно проделать всю работу заново. Вывод, правильность которого выяснится через месяц.

Почему это критично. Агент работает без вас. Если ошибку нельзя быстро заметить, она накопится — и вы узнаете о ней от клиента, как в примере из четвёртой главы.

4. Ошибка обратима

Что произойдёт, если агент отработает неправильно и вы не заметите?

Обратимо: неверная строка в таблице, неудачный черновик, лишнее уведомление. Заметили — поправили.

Необратимо: отправленное клиенту письмо, проведённый платёж, удалённые данные, публикация.

Это не запрет, а требование к схеме. Задачу с необратимым результатом можно автоматизировать — но последним шагом должно быть ваше подтверждение. Агент готовит, человек нажимает. Тогда критерий выполняется: до вашего нажатия всё обратимо.

Что реально работает

По практике рынка список коротких и надёжных сценариев выглядит так — и он на удивление скучный.

Обработка входящих. Заявки, письма, сообщения. Работает потому, что поток однотипный: в любом деле большая часть обращений повторяется. Разложить по типам, записать, подготовить черновик ответа.

Ответы на типовые вопросы. Первая линия — самый зрелый и проверенный сценарий из всех. Основная масса вопросов у любого бизнеса одинакова.

Документооборот. Извлечь данные из счёта, накладной, договора и положить в таблицу. Скучно, регулярно, легко проверяется.

Подготовка контента. Черновики по расписанию, описания, адаптация текста под площадки. Важно: черновики, а не публикация.

Сбор и сведение данных. Взять из трёх мест, соединить, посчитать формулой, прислать сводку.

Обратите внимание на общее. Все пять — про перекладывание и разбор, а не про решения. В каждом есть чёткий вход и чёткий выход. Ни в одном агент ничего не решает.

Что не работает

Задачи с решениями. Кому дать скидку, брать ли заказ, что ответить недовольному клиенту в нестандартной ситуации. Агент выдаст усреднённый разумный ответ, не зная вашего контекста.

Переговоры и живое общение. Всё, где на той стороне человек, реагирующий на реакцию.

Разовые задачи. Даже большие. Проще сделать руками.

Задачи, которые вы делаете по-разному. Пока не стандартизировали — автоматизировать нечего.

Задачи, где вы не знаете, как делаете. Если работа опирается на опыт и чутьё, она не раскладывается на шаги.

Творческая работа целиком. Отдельные куски — да, целиком — нет.

Главная ошибка: слишком большая задача

Самая частая причина провала, и она случается на этом этапе.

Как это выглядит. Вместо «раскладывать входящие заявки по типам» человек берётся за «автоматизировать работу с клиентами». Вместо одного узкого процесса — целое направление.

Что происходит дальше. Агент, который делает десять разных вещей, путается, отвечает медленно и плохо тестируется. Поменяли формулировку в одной задаче — посыпались остальные. Через два месяца это либо бросают, либо оно портит работу.

Что говорит арифметика. Узкий процесс с высокой частотой окупается за недели или пару месяцев. Глобальный «ИИ-отдел» — за годы или никогда.

Правило:

Несколько узких агентов лучше одного универсального.

Узкий агент собирается за вечер, ломается предсказуемо и чинится за десять минут. Универсальный собирается месяц и ломается непонятно.

Практический признак правильного размера. Задачу можно описать одним предложением без слова «и». «Раскладывать входящие заявки по типам и записывать в таблицу» — это одно предложение с «и», но одна задача: заявки на входе, строки на выходе. А вот «раскладывать заявки и отвечать клиентам и вести отчётность» — три агента.

Как выбрать первую

Практическая часть главы. Пятнадцать минут работы.

Шаг 1. Выпишите повторяющиеся задачи. Всё, что делаете регулярно и что связано с перекладыванием, разбором или подготовкой текста. Не думайте пока, годится или нет.

Шаг 2. Проставьте частоту и время. Сколько раз в неделю и сколько минут занимает один раз. Умножьте — получите недельные потери.

Шаг 3. Прогоните по четырём критериям. Отсеките всё, где не выполняется хотя бы одно условие.

Шаг 4. Из оставшегося выберите не самую крупную, а самую простую.

Вот здесь читатели обычно ошибаются. Соблазн — начать с задачи, которая съедает больше всего времени. Но первый агент — это ещё и обучение: вы будете разбираться с инструментом, спотыкаться и переделывать. На простой задаче вы дойдёте до конца, на сложной бросите на середине.

Правильный первый агент выглядит так: пять-семь шагов, один умный, понятный вход, понятный выход, результат приходит вам, а не наружу.

Экономит он, может быть, двадцать минут в неделю. Это неважно. Важно, что после него вы соберёте второй за вечер — а второй возьмёте уже тот, который экономит два часа.

Таблица для отбора

Заполните по своим задачам — это и есть результат главы.

| Задача | Раз в неделю | Минут за раз | Повторяется | Шаги одинаковы | Результат проверяем | Ошибка обратима | Годится |

|---|---|---|---|---|---|---|

| Разбор входящих заявок | 40 | 3 | да | да | да | да | **да** |

| Утренний разбор почты | 5 | 20 | да | да | да | да | **да** |

| Ответы на типовые вопросы | 30 | 4 | да | да | да | да (с подтверждением) | **да** |

| Отчёт для руководителя | 1 | 60 | да | да | да | да | **да** |

| Решение по скидкам | 10 | 5 | да | **нет** | нет | нет | нет |

| Квартальная сверка | 0,1 | 180 | **нет** | да | да | да | нет |

Четыре первые строки — кандидаты. Из них для первого агента берите не отчёт — он длиннее и уходит наружу, — а что-то из ежедневного: разбор заявок или утренний разбор почты. Чаще, проще, ошибка дешевле.

Дальше в книге сквозным примером идёт разбор почты: в следующей главе мы разложим его на шаги, в четвёртой посмотрим, где он ломается, в пятой посчитаем окупаемость, а в седьмой соберём. Если у вас своя задача — держите её рядом и сверяйте.

Где эта глава не поможет

Если у вас нет повторяющихся задач. Бывает при проектной работе, где каждый заказ новый. Тогда агенты вам, скорее всего, не нужны — а нужны хорошие запросы, и это другая книга.

Если задача одна и она сложная. Иногда единственная стоящая автоматизации задача не проходит по критериям. Честный ответ: не автоматизировать. Плохой агент хуже отсутствующего.

Если вы не считали время. Все критерии опираются на то, что вы знаете, сколько тратите. Если не знаете — неделю позаписывайте, прежде чем выбирать.

Итог главы

Первая мысль о задаче обычно неверная: раздражает то, что требует решений, а именно это не отдаётся агенту.

Четыре критерия, и нужны все: повторяется, шаги одинаковы, результат проверяем, ошибка обратима.

Частота важна, но количество событий важнее: двадцать раз в день по минуте окупится быстрее, чем раз в неделю по часу.

Проверка на одинаковость шагов: можете ли описать эту работу новому человеку без слов «тут по ситуации».

Необратимый результат — не запрет, а требование: последним шагом ваше подтверждение.

Что работает: обработка входящих, ответы на типовые вопросы, документооборот, черновики контента, сведение данных. Всё скучное и без решений.

Что не работает: решения, переговоры, разовое, нестандартизированное, опирающееся на чутьё.

Главная ошибка — слишком большая задача. Узкий процесс окупается за недели, глобальный — за годы или никогда.

Несколько узких агентов лучше одного универсального.

Первым берите не самую крупную задачу, а самую простую: первый агент — это обучение.

Что сделать сегодня. Заполните таблицу отбора по своим задачам. Пятнадцать минут. Выберите одну — простую, частую, с результатом, который приходит вам, а не клиенту. Она и будет вашим первым агентом, а следующая глава научит раскладывать её на шаги.

Следующая глава — главная в книге.

Глава 3. Разложить задачу на шаги

Ситуация

Вы выбрали задачу. Она проходит по всем четырём критериям из прошлой главы: повторяется, шаги одинаковы, результат проверяем, ошибка обратима.

Вы открываете конструктор. Перед вами пустое поле и кнопка «добавить шаг».

И вы обнаруживаете, что не знаете, какой шаг первый.

В голове задача выглядит цельной: «разбирать входящие заявки». Одно действие. Но конструктор просит разложить его на части, а части не раскладываются — потому что вы никогда не думали об этой работе как о последовательности. Вы просто её делали.

Дальше обычно происходит одно из двух. Человек бросает, решив, что это не для него. Или собирает агента из трёх шагов, в среднем из которых написано «модель разберётся», — и он не работает.

Эта глава — про навык, который решает всё остальное. Он не про технологии: раскладывать задачи на шаги полезно и без всяких агентов.

Почему это главное

Формулировка, ради которой написана глава:

Инструмент вторичен. Раскладка первична.

Человек, который умеет разложить задачу, соберёт агента в любом конструкторе — за вечер, разбираясь по ходу. Человек, который не умеет, не соберёт ни в каком, сколько бы уроков ни посмотрел.

Причина простая: **конструктор не придумывает шаги, он их соединяет**. Все инструменты, которые вы встретите, — это способы записать последовательность, которую вы уже придумали. Придумывать её они не умеют.

Из этого следует практический порядок работы:

Сначала бумага, потом экран. Раскладка делается до открытия конструктора. Иначе вы будете придумывать логику, одновременно борясь с незнакомым интерфейсом, и провалите обе задачи.

Правило одного глагола

Базовое правило, из которого растёт всё остальное.

Один шаг — один глагол.

Взять. Проверить. Разложить. Записать. Отправить. Посчитать. Сравнить.

Если для описания шага вам нужно два глагола — это два шага. «Взять письмо и записать в таблицу» — два шага, даже если вам кажется, что действие одно.

Зачем такая строгость. Три причины, и все практические.

Первая: шаг с двумя глаголами нельзя проверить. Если он отработал неправильно, непонятно, на какой половине сломалось.

Вторая: его нельзя переиспользовать. Разбив на два, вы можете вставить между ними проверку, а потом заменить одну половину, не трогая другую.

Третья: пока вы не разбили, вы не думали. Формулировка «взять и записать» скрывает вопросы: в каком виде взять, что делать, если поля пустые, куда именно записать, что если строка уже есть. Разбив, вы натываетесь на эти вопросы — а наткаться лучше на бумаге, чем на третьей неделе работы агента.

Вход и выход каждого шага

Второе правило, которое отличает работающую раскладку от списка пожеланий.

У каждого шага должно быть написано, что приходит на вход и что получается на выходе.

Пример плохой раскладки:

Взять письмо

Понять, о чём оно

Ответить

Пример той же задачи в рабочем виде:

Взять письмо. Вход: новое письмо в папке «Входящие». Выход: текст письма, адрес отправителя, тема, дата.

Определить тип. Вход: текст письма. Выход: одно слово из списка — заявка, вопрос, жалоба, спам.

Составить черновик. Вход: текст письма, тип, мой шаблон для этого типа. Выход: текст ответа.

Отправить черновик мне. Вход: текст ответа, адрес отправителя. Выход: сообщение в мессенджере.

Разница не в подробности. Разница в том, что во втором варианте видно, откуда что берётся, — и сразу заметно, если чего-то не хватает.

Практический приём: пройдите по своей раскладке снизу вверх и для каждого шага спросите — откуда я возьму то, что мне здесь нужно? Если ответа нет, значит, пропущен шаг.

Развилки

Третье правило, и его пропускают чаще всего.

Всё, что может пойти не так, должно быть на схеме.

Агент не сообразит, что делать в необычной ситуации. Он выполнит записанный шаг с любыми данными, включая те, для которых шаг не предназначен.

Четыре развилки, которые есть почти в каждой задаче:

Данных нет. Письмо без вложения. Пустое поле в форме. Таблица без новых строк. Что делать: пропустить, взять значение по умолчанию, сообщить вам?

Данные не такие. В поле «сумма» написано «уточним позже». Файл не того формата. Текст на другом языке. Это самая частая причина поломок.

Ответ модели неожиданный. Вы просили одно слово из четырёх, а пришёл абзац рассуждений. Или слово не из списка.

Внешний сервис не ответил. Почта недоступна, таблица не открылась, сервис перегружен. Повторить попытку? Через сколько? Сколько раз?

Как это записывается. Не отдельным сложным механизмом, а просто пунктом в раскладке:

2. Определить тип. Вход: текст письма. Выход: одно слово из списка. **Если пришло не слово из списка** — записать «неопределённый» и отправить письмо мне целиком.

Правило для первого агента: любая развилка, которую вы не знаете, как обработать, ведёт к вам. Сообщение в мессенджер с текстом «не смог обработать, посмотри». Это не изящно, зато вы узнаете о проблеме сразу, а не через месяц.

Главная ошибка: шаг «модель разберётся»

Самая частая причина, по которой первый агент не работает.

Как это выглядит. В раскладке появляется шаг вроде:

2. Обработать заявку

Или:

3. Разобраться с письмом и сделать что нужно

Это не шаг. Это описание всей задачи, вставленное в середину списка.

Почему так получается. Вы знаете, что модель умная, и подсознательно ожидаете, что она поймёт задачу целиком, как поняли бы вы. В чате это отчасти работает — там вы рядом и поправите. В агенте вас рядом нет.

Как это чинится. Задайте себе один вопрос:

Что конкретно должно оказаться на выходе этого шага? В каком виде?

Если ответ «ну, обработанная заявка» — шаг не разложен. Если ответ «одно слово: срочная, обычная или не по адресу» — разложен.

Проверка на «умный шаг». У правильно разложенного умного шага всегда есть три вещи: конкретный вопрос к модели, конкретный формат ответа и список того, чего в ответе быть не должно. Если хоть одного нет — вернитесь и доразложите.

Как раскладывать, если не получается

Практический метод для тех, кто смотрит на задачу и не видит в ней шагов.

Способ 1. Проговорить вслух

Опишите вслух, что вы делаете, когда выполняете эту задачу руками. Не «что должно происходить», а что делаете вы: открываю почту, смотрю на отправителя, если это клиент — открываю письмо, читаю первые строки, понимаю, о чём...

Каждое «открываю», «смотрю», «читаю», «понимаю» — кандидат в шаги.

Работает потому, что вы не помните процесс целиком, но прекрасно помните действия, когда их проговариваете.

Способ 2. Интервью с самим собой

Приём из книги «Нейросети для работы», здесь особенно уместный:

Мне нужно разложить на шаги процесс: [описание задачи одной фразой]. Я делаю это руками и знаю, как это работает, но не могу описать по шагам.

Задавай мне вопросы по одному: что я делаю первым, откуда беру данные, как принимаю решение на каждой развилке, что происходит с необычными случаями, чем всё заканчивается.

Не составляй раскладку, пока не соберёшь всё. Спрашивай про то, что обычно забывают: пустые данные, ошибки, редкие случаи.

Пятнадцать минут ответов — и раскладка собирается сама.

Важно: последнее предложение обязательно. Именно редкие случаи вы забудете, а именно они сломают агента.

Способ 3. Пойти от результата

Начните с конца: что должно получиться? Дальше — что нужно, чтобы это получилось? И так до самого начала.

Работает лучше движения вперёд, когда задача сложная: двигаясь назад, вы не можете пропустить шаг, потому что каждый следующий вопрос — «а это откуда возьмётся».

Разбор: одна задача, две раскладки

Возьмём ту задачу, которую вы, скорее всего, выбрали в прошлой главе, — разбор утренней почты. Каждое утро вы просматриваете накопившиеся письма и решаете, на что отвечать сегодня, что подождёт, а что можно не открывать.

Это же задача, которую мы будем собирать в седьмой главе. Сейчас — только раскладка.

Раскладка первая, плохая

Взять письма

Разобрать почту

Прислать мне

Что не так. Второй шаг — вся задача целиком. Ни одного входа и выхода. Ни одной развилки.

Такой агент соберётся за десять минут и не будет работать: модель получит гору писем и просьбу «разобрать почту», а вернёт каждый раз что-то новое — то список, то рассуждение, то пересказ трёх первых писем.

Раскладка вторая, рабочая

Взять непрочитанные письма за сутки. Вход: почтовый ящик, вчерашняя дата. Выход: список писем — отправитель, тема, первые строки.

Если писем нет — отправить «за сутки писем не было» и остановиться.

Проверить, что письма пришли в ожидаемом виде. Вход: список. Выход: годится или нет.

Полей не хватает — сообщить мне и остановиться.

Отсеять рассылки. Вход: список. Выход: список без рассылок.

Механически: по наличию ссылки отписки и по отправителям из моего списка. Без модели.

Определить срочность каждого письма. Вход: отправитель, тема, начало текста. Выход: одно слово из трёх — сегодня, подождёт, не требует.

Ограничения для модели: ровно одно слово из списка; никаких пояснений и знаков препинания.

Проверить ответ. Вход: слово. Выход: годится или нет.

Слово не из списка — пометить письмо как «сегодня».

Собрать сводку. Вход: письма с метками. Выход: текст: сначала срочные, потом остальные, в конце — сколько отсеяно рассылок.

Проверить длину. Вход: текст. Выход: годится или нет.

Длиннее лимита мессенджера — оставить только срочные, остальные свернуть в строку «и ещё столько-то».

Отправить мне в мессенджер. Вход: текст. Выход: сообщение.

Что изменилось.

Четыре механических шага, один умный, три проверочных.

Проверочных здесь необычно много — три на восемь шагов. Так и должно быть у агента, который работает без вас.

Третий шаг — механический отсев до обращения к модели. Без него агент отправляет модели все сорок писем, включая двадцать рассылок, и половина денег уходит впустую.

Четвёртый шаг сформулирован так, что его результат можно проверить механически: одно слово из известного списка. Именно поэтому возможен пятый шаг. Попроси мы «оценить важность письма», проверить было бы нечего.

Есть три развилки: писем нет, данные не те, ответ модели не из списка. Первые две ведут к вам, третья — к безопасному варианту: лучше лишний раз побеспокоить, чем пропустить важное.

Седьмого шага в первом черновике не было. Он появился, когда я пошёл по раскладке от результата назад и спросил: а что придёт мне, если писем окажется шестьдесят? Так и находятся пропущенные шаги.

Время на раскладку: двадцать минут на бумаге. Время на сборку по готовой раскладке: вечер. Время на сборку без раскладки: неделя, и результат хуже.

Второй пример: там, где считают

Задача другого типа — чтобы видно было, что метод не про почту.

Раз в неделю вы собираете отчёт по продажам: выгружаете данные из таблицы, считаете итоги, пишете короткий комментарий, отправляете руководителю.

Раскладка:

Взять строки за прошедшую неделю. Вход: таблица продаж, дата. Выход: набор строк.

Строк нет — сообщить и остановиться.

Посчитать итоги формулой. Вход: строки. Выход: сумма, количество сделок, средний чек, разбивка по направлениям.

Считает таблица, не модель.

Взять итоги прошлой недели. Вход: лист истории. Выход: те же числа за прошлый период.

Посчитать изменения формулой. Вход: две группы чисел. Выход: изменения в процентах.

Написать комментарий. Вход: итоги, изменения, мой шаблон. Выход: три абзаца.

Ограничения: не добавлять чисел, которых нет во входных данных; не строить прогнозов; до 900 знаков.

Проверить текст. Вход: комментарий, исходные числа. Выход: годится или нет.

Есть числа, которых нет в данных, — отправить мне на проверку вместо отправки руководителю.

Собрать письмо. Вход: комментарий, таблица итогов. Выход: письмо.

Отправить. Вход: письмо, адрес. Выход: отправленное письмо и копия мне.

Записать итоги в лист истории. Вход: числа, дата. Выход: новая строка.

Что здесь важно.

Все расчёты делает таблица. Модель только формулирует — она не считает, а генерирует правдоподобное на месте вычисления. В агенте это критичнее, чем в чате: в чате вы бы заметили странное число, а здесь письмо уйдёт само.

Девятый шаг нужен для третьего шага на следующей неделе. Двигаясь вперёд, его легко пропустить; двигаясь от результата назад — находишь сразу.

И обратите внимание на восьмой шаг: письмо уходит руководителю. Значит, шестая проверка обязательна — это то самое «действие, которое видит кто-то кроме вас» из чек-листа ниже.

Чек-лист готовой раскладки

Прежде чем открывать конструктор, проверьте:

—

☐ У каждого шага один глагол

—

☐ У каждого шага записано, что на входе и что на выходе

—

☐ Для каждого входа понятно, откуда он берётся

—

☐ Умных шагов один-два, в сложном случае три; для каждого указан формат ответа

—

☐ Нет шагов вида «обработать», «разобраться», «сделать»

—

☐ Все расчёты делает не модель

—

☐ Есть хотя бы одна проверка перед действием, которое видит кто-то кроме вас

—

☐ Для каждой развилки написано, что делать; неизвестные случаи ведут к вам

—

☐ Понятно, куда попадёт результат и кто его увидит

—

☐ Если результат уходит наружу — есть шаг с вашим подтверждением

Если хотя бы один пункт не отмечен, возвращайтесь к бумаге. Это дешевле, чем обнаружить пропуск после сборки.

Где это не сработает

Задачи с неявными решениями. Есть работа, где вы принимаете решения на основании того, что не формулируется: опыта, чутья, знания конкретных людей. Такая задача не раскладывается — и это не ваша неспособность, а свойство задачи.

Слишком длинные цепочки. Если раскладка вышла на двадцать пять шагов, задача велика. Разделите на два агента: один готовит данные, второй работает с подготовленным. Почему длинные цепочки плохи — следующая глава.

Процессы, которые меняются каждый раз.

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.