

Oper8

AI-процессы, которые учатся с каждым клиентом



Сергей Липчанский · Асхат Уразбаев

Содержание

Орег8: строим ИИ-процессы, которые учатся с каждым клиентом	5
Об авторах	5
Навигатор: у вас проблема, вот глава	6
Благодарности	6
Часть 1. Что меняется	7
Четыре сдвига: что становится возможным	8
Тот же день, другой сценарий	8
Один клиент	9
Каждый контакт учит	9
Почему конкурент не догонит	10
Мгновенность	10
Ловушка Deploy: почему 74% компаний не видят окупаемость от ИИ	11
Deploy-версия «ФудПро»	11
Четыре сдвига, которых Deploy не даёт	12
Парадокс 80/80	13
Как понять, где вы	13
Эта книга про Reshape	14
Процесс, который учится	14
Два цикла на одном примере	15
Маховик данных	16
Не каждый процесс — кандидат	16
Владелец процесса и знание	17
Человек у руля	17
Знание, которое не уходит с людьми	18
Слон в комнате	18
Выбор: что менять и насколько глубоко	19
Три кандидата	20
Три барьера	20

Насколько глубоко?	21
Две ловушки выбора	21
Две мысли на дороге	22
Часть 2. Маховик	23
Как устроен маховик	24
База знаний: откуда ИИ знает, как работать	24
Три слоя	26
Как собирать	27
Мёртвая база знаний	27
Автономия: сколько свободы у ИИ и где границы	28
Лестница: от подсказчика до автопилота	28
Три слоя правил	29
Как переходить безопасно	30
Откат — штатная операция	31
Границы живые	32
Память: что решили, почему и оказалось ли верным	33
Что записывать	33
Три уровня зрелости	34
Петля без обратной связи	35
Проверка качества: работает ли и стало ли лучше	36
Как устроен тест	36
Версионирование	37
Две ошибки	37
Метрики: процесс улучшается?	38
Четыре метрики	39
Контрметрика: зачем нужен противовес	40
Замыкание маховика	40
Часть 3. Масштаб	42
Когда среднее звено тормозит	43

Как это выглядит	43
Как ФудПро разморозил Игоря	44
Пять рычагов	45
ИИ-грамотность: четыре уровня	46
Четыре уровня	46
Что не работает	47
Обучить и забыть	48
Роли: кто за что отвечает	48
Три роли	49
Старые роли не исчезают	50
Карта перехода	50
Часть 4. Старт	52
Что делать в понедельник	53
Четыре шага	53
Когда что-то идёт не так	54
Что дальше	54
От теории к практике	56

Орег8: строим ИИ-процессы, которые учатся с каждым клиентом

Большинство компаний внедряют ИИ одинаково: берут старый процесс и прикручивают к нему чат-бот, помощник или автоматизацию. Получается быстрее, но не принципиально лучше. По данным McKinsey, три из четырёх организаций не видят от этого окупаемости.

Эта книга про другой путь. Не ускорять старое, а перестроить процесс так, чтобы он становился умнее с каждым клиентом, с каждым контактом, с каждым днём. Мы называем это *маховик данных*, и на протяжении книги соберём его по частям.

Об авторах

У Сергея Липчанского и Асхата Уразбаева есть профессиональная деформация: как только в индустрии что-то устаканивается и все выдыхают, эти двое находят следующую амбразуру и бегут её закрывать.

Пятнадцать лет назад они объясняли руководителям, что привычный способ управлять проектами устарел, а командам нужно больше свободы и меньше согласований. Руководители смотрели с подозрением. Потом те же руководители перестроились и сделали вид, что всегда так думали.

Теперь история повторяется: только вместо перестройки управления теперь перестройка процессов с ИИ. Реакция руководителей пока примерно та же.

Сергей Липчанский, основатель Кактус.ИИ, операционный директор ScrumTrek. Двадцать лет ломает компаниям привычный способ работы и удивляется, что его продолжают приглашать. [Написать Сергею](#)

Асхат Уразбаев, основатель ScrumTrek. Когда ему стало скучно перестраивать управление, он переключился на ИИ-трансформации целых операционных моделей. Масштаб амбразуры вырос, привычка бежать на неё осталась. [Написать Асхату](#)

Навигатор: у вас проблема, вот глава

Ваша ситуация	Куда читать
«Не понимаем, зачем нам ИИ»	1.1 <i>Четыре сдвига</i>
«Запустили пилоты, но окупаемости нет»	1.2 <i>Ловушка Deploy</i>
«Не понимаем, какой процесс менять первым»	1.5 <i>Выбор: что менять и насколько глубоко</i>
«ИИ галлюцинирует, не знает наши правила»	2.1 <i>База знаний</i>
«Боимся дать ИИ больше свободы»	2.2 <i>Автономия и рулбук</i>
«ИИ принимает решения, но мы не знаем какие»	2.3 <i>Память: фиксация решений</i>
«Качество упало после обновления»	2.4 <i>Гейт качества</i>
«Не можем доказать бизнесу, что ИИ окупается»	2.5 <i>Метрики</i>
«Средний менеджмент саботирует внедрение»	3.1 <i>Когда среднее звено тормозит</i>
«Люди не умеют работать с ИИ»	3.2 <i>ИИ-грамотность</i>
«Не понимаем, кто за что отвечает»	3.3 <i>Роли</i>

Благодарности

Спасибо людям, которые терпеливо вычитывали черновики и не давали авторам улететь в стратосферу: Ольга Качалина, Мария Петрова, Олеся Якубова.

И ИИ-агентам, которые работали над текстом без выходных, перерывов на кофе и претензий по зарплате: Аскар, Кира, Гай, Марат, Игорь.

Мы готовим серию практических гайдов с чек-листами и шаблонами. Подписывайтесь, чтобы не пропустить: t.me/aidea4work

Часть 1. Что меняется

Новый взгляд на процессы, продукт и знание

Четыре сдвига: что становится возможным

Два месяца назад из «ФудПро» уволился Дима, один из лучших менеджеров по продажам. «ФудПро» поставляет продукты в рестораны и кафе. Четыреста клиентов в городе. Текучка в отделе продаж обычная для отрасли: за год меняется треть команды.

Среди Диминых клиентов — ресторан «Тополь», который приносил стабильные заказы два года.

Новый менеджер Лена получает Димины чаты и папку с пометками. Пишет шеф-повару «Тополя», представляется. «А Дима где?» Ушёл. Лена отправляет подборку текущих акций. Шеф читает, но не отвечает.

Лена не знает, что «Тополь» каждый четверг заказывает фермерскую птицу под выходной поток гостей, что в марте ресторан меняет меню и за две недели до этого нужно предложить сезонные позиции, что этому шефу нельзя продавать через скидки и давление на объём — с ним работает другое: дать попробовать, рассказать про продукт, подождать.

Всё это Дима знал. Но знал головой, не системой. Через неделю шеф берёт пробную поставку у конкурента.

Тот же день, другой сценарий

Дима увольняется, и ничего не обнуляется.

В четверг утром ИИ-агент пишет шефу «Тополя» в привычный мессенджер: «Фермерская птица на выходные: партия от “Угличе Поле”, охлаждённая, поступила вчера. Как обычно 8 кг?» Шеф подтверждает одним сообщением. Он может даже не знать, что Дима ушёл.

Агент знает три вещи.

1. **История клиента.** Ритм заказов (птица по четвергам, рыба нерегулярно, сезонная закупка в начале марта и сентября), предпочтения шефа (фермерские поставщики, мелкая фасовка), что последний раз сработало, а что нет.
2. **Правила работы** с этим типом клиента. Не давить на объём. Предложить попробовать новую позицию бесплатно. Перед сменой меню связаться за две недели, а не за два дня.
3. **Тон.** С этим шефом разговор строится вокруг качества продукта, не цены. Спокойно, предметно, без «суперакции только до пятницы».

А Лена? Лена больше не менеджер по продажам. Она разбирает случаи, которые агент не вытянул: конфликт с доставкой, потерянный клиент, нестандартный запрос. Она же калибрует правила, по которым агент работает: смотрит, какие подходы дали результат, какие нет, и обновляет их. Двадцать менеджеров обслуживали четыреста клиентов. Теперь четыреста клиентов ведёт агент, а пять человек делают его умнее.

Что за этим стоит? Четыре сдвига, каждый из которых стал возможен только сейчас.

Один клиент

ИИ Агент писал не «ресторану среднего сегмента в центре», а конкретному «Тополю» и его конкретному шефу. Компания перестала думать сегментами.

Раньше индивидуальный подход означал опытного менеджера, который помнит каждого из пятидесяти клиентов. Это работает, пока человек не уволится. Поэтому большинство компаний работает по сегментам: стандартная акция для всех ресторанов, типовая рассылка. Нормально, но одинаково.

Когда ИИ анализирует контекст за копейки, индивидуальный подход перестаёт зависеть от памяти конкретного сотрудника. «ФудПро» строит отношения с каждым из четырёхсот клиентов отдельно и не теряет это при смене команды. Любой бизнес, где клиентов больше, чем способен помнить один человек, оказывается в той же ситуации: поставщик стройматериалов, логистическая компания, розничная сеть.

Каждый контакт учит

Откуда агент знал, что шефу нельзя продавать через скидки и что сезонную закупку нужно предлагать за две недели?

Из опыта. Каждый заказ, каждый отказ, каждое прочитанное и не отвеченное сообщение становится обучающим сигналом.

Когда Дима два года назад попробовал предложить «Тополю» скидку за объём, шеф неделю не отвечал на сообщения. Это стало правилом: с данным типом клиента скидка за объём вредит. Когда другой менеджер отправил бесплатный образец похожему ресторану, заказ вырос на 20%. Это тоже стало правилом.

Но учится не только знание о клиенте. Учатся сами правила работы: какой подход к удержанию даёт результат, как лучше писать, когда связываться, что предлагать. Лучшие ходы лучших менеджеров становятся стандартом для всей компании, а не уходят вместе с

человеком.

У «ФудПро» четыреста клиентов. Это четыреста параллельных потоков обучения, где каждый удачный ход и каждая ошибка формализуются. Через год система знает про работу с ресторанами больше, чем лучший менеджер в команде, потому что ни один менеджер не видел четыреста историй отношений одновременно. И это знание не обнуляется, когда треть команды меняется за год.

Почему конкурент не догонит

На рынок выходит новый поставщик. Свежий склад, те же бренды, чуть ниже цены. Купил ту же платформу. Почему клиент останется в «ФудПро»?

Потому что у «ФудПро» два года данных. Какие подходы работают с какими клиентами. Какие рестораны начинают уходить после определённых ошибок (и как их вернуть). Какой тон общения даёт лучший результат с разными типами шефов. Тысячи мелких решений, отточенных реальной обратной связью. Не просто данные о заказах, а знание о том, как строить отношения и как делать это с каждым месяцем лучше.

Новый поставщик может выставить цены ниже. Но два года накопленных решений так легко не скопируешь. Каждый месяц работы делает систему точнее, и разрыв с конкурентом растёт.

Создавать просто продукт (платформу, каталог, приложение для заказа) бессмысленно: навайбкодят копию за месяц. Защищает только сервис, собранный из миллиона мелких решений, которые копить долго, а повторить невозможно. Это работает одинаково для поставщика продуктов, для сети стройматериалов, для страховой компании. ИИ впервые позволяет доставлять такой сервис в масштабе, без линейного роста штата.

Мгновенность

Одиннадцать вечера. Шеф-повар «Тополя» понимает, что завтра банкет на сорок человек, а лосося не хватит. Раньше надо было ждать утра, писать менеджеру, менеджер проверяет склад, пишет обратно. Полдня до ответа, и не факт, что успеют довести.

Теперь шеф пишет в чат «ФудПро». Ответ через десять секунд: «Лосось охлаждённый, 12 кг на складе. Вам обычно нужна норвежская фасовка 1,5 кг. Доставка завтра к 8 утра, 28 400 р. Подтвердить?» Шеф подтверждает и идёт спать.

Мгновенность меняет не только удобство, но и поведение. Человек, который получает ответ за секунды, обращается с вопросами, которые раньше не задавал («не буду дёргать

менеджера из-за мелочи»). Но эта мелочь часто оказывается началом нового заказа. Мгновенный доступ не просто экономит время. Он увеличивает количество точек контакта, а каждая точка контакта учит систему.

Все описанное — результат системы, которую мы будем собирать на протяжении книги. Внутри неё крутится механизм из пяти шестерёнок, *маховик данных*. Но прежде чем его собирать, нужно разобраться с ловушкой, в которую попадают три из четырёх компаний, пытающихся внедрить ИИ.

Ловушка Deploy: почему 74% компаний не видят окупаемость от ИИ

В прошлой главе мы видели, как «ФудПро» перестроил работу с клиентами. ИИ-агент ведёт четыреста ресторанов, Лена калибрует правила, знания не теряются при текучке, шеф-повар получает ответ в одиннадцать вечера. Красивая картинка.

Но «ФудПро» мог пойти другим путём. Большинство компаний именно так и делает.

Deploy-версия «ФудПро»

Дима увольняется. Руководство покупает «умную CRM» с интеллектуальным ИИ-модулем, который умеет отвечать на вопросы по документам. Менеджеры получают подсказки: история заказов клиента, рекомендация «позвонить» или «предложить акцию», средний чек, дата последнего контакта.

Лена открывает карточку «Тополя» и видит: последний заказ две недели назад, средний чек 45 тысяч, рекомендация «предложить сезонные позиции». Вроде полезно. Лена пишет шефу, предлагает сезонное меню. Шеф читает, не отвечает. Лена не понимает почему.

CRM не знает, что этому шефу нельзя продавать через ассортимент — ему важно потрогать, попробовать, составить собственное мнение. Это знал Дима. CRM хранит историю заказов, но не историю отношений.

Через месяц руководство смотрит отчёт. Менеджеры тратят на каждого клиента на 15% меньше времени. Неплохо. Но клиентов больше не стало, средний чек не вырос, а три

ресторана тихо ушли к конкуренту. «Умная CRM» помогла делать то же самое чуть быстрее. Не более.

Компания Boston Consulting Group называет такой подход к внедрению Deploy (внедрение): добавить ИИ к тому, что уже работает. Тот же процесс, те же роли, те же решения, просто с подсказкой на экране. Айтишники называют это *raving the cowpath*, асфальтирование коровьей тропы. Тропа стала ровнее, но маршрут по-прежнему идёт через болото.

Deploy даёт +10-20% эффективности. По данным McKinsey (2024), 74% организаций не видят значимой окупаемости от ИИ. Большинство из них застряли именно здесь.

Четыре сдвига, которых Deploy не даёт

В прошлой главе мы разобрали четыре сдвига. Каждый из них требует другого процесса, а не ускорения старого. Вот почему Deploy упирается в потолок.

Мгновенность. Deploy ускоряет человека, но человек не ответит шефу в одиннадцать вечера за десять секунд. А если ответит, завтра ему напишут ещё двадцать клиентов. Больше точек контакта при Deploy означает больше нагрузки на людей. Линейно. Чтобы отвечать мгновенно и в масштабе, нужен агент. А агент, который сам отвечает клиентам, это уже не подсказка в CRM. Это другой процесс.

Каждый контакт учит. В Deploy-версии CRM копятся записи: даты, суммы, заказы. Но записи не становятся правилами. Система видит, что «Тополь» заказывал птицу по четвергам. Не видит, почему один подход к шефу работает, а другой разрушает отношения. Данные копятся, но не превращаются в знание. Петля обучения не замыкается.

Один клиент. Лена видит подсказки CRM, но всё равно пишет по скрипту: одна акция на всех, одно сообщение на весь сегмент «рестораны, центр, средний чек выше 40 тыс». Индивидуальный подход к четырёмстам клиентам при Deploy означает нанять ещё двадцать менеджеров, каждый из которых запомнит своих пятьдесят. А потом они уволятся, и цикл начнётся заново.

Сервис как защита. Через дорогу открывается конкурент. Покупает ту же CRM, нанимает менеджеров, даёт цены чуть ниже. И оказывается в той же точке, что и «ФудПро». Потому что у Deploy-версии нечего защищать: нет накопленного знания, нет отточенных правил, нет обучившегося агента. Только подписка на платформу, которую может купить кто угодно.

Вот четыре причины, по которым Deploy упирается в потолок. Не потому что ИИ “делает какую-то ерунду”. Потому что подход не позволяет использовать то, что ИИ на самом деле умеет.

Парадокс 80/80

По данным Boston Consulting Group, 80% реальной окупаемости от ИИ лежит в перестройке процессов. Но 80% корпоративных бюджетов уходит на Deploy: лицензии, интеграции, пилоты, которые красиво смотрятся на квартальном отчёте.

Почему? Потому что Deploy понятен. Купил лицензию, подключил, показал, что менеджеры работают на 15% быстрее. Перестройка процессов это страшно: другие роли, другие данные, другие метрики, люди сопротивляются, результат не завтра. Проще прикрутить подсказку к тому, что есть, и на этом успокоиться.

Перестройку мы в этой книге называем *Reshape* (перестраивание). Это когда вы не прикручиваете ИИ к старому процессу, а проектируете процесс заново, исходя из того, что ИИ умеет.

Перестраивается все:

- Роли. Лена калибрует правила, а не тупо пишет по скрипту.
- Данные. Каждый контакт порождает обучающий сигнал, а не просто запись в CRM.
- Маршрут. Агент сам ведёт клиента, человек подключается в сложных случаях.

Как понять, где вы

Три вопроса помогают определить, в каком подходе вы находитесь.

1. **Роли.** Изменилась ли функция людей после внедрения ИИ? Если менеджер делает то же самое, только быстрее, вы в Deploy. Если менеджер перестал писать клиентам и начал проектировать правила для агента, это Reshape. Простая проверка: откройте описание должности через полгода после запуска. Если оно не изменилось, процесс не перестроен.
2. **Данные.** Появились ли данные, которых раньше не было? Deploy использует те же данные эффективнее. Reshape порождает новые: лог решений с обоснованием, обратная связь по каждому контакту, правила, которые формализуются из опыта. Если ваш ИИ-процесс не производит данные, которые улучшают следующий прогон, маховик не крутится.
3. **Маршрут.** Может ли ИИ вести клиента без человека-посредника на каждом шаге? В Deploy человек остаётся обязательным звеном: получил подсказку, принял решение,

написал клиенту. В Reshape агент ведёт рутину сам, а человек подключается там, где нужен человек.

Два из трёх положительно — перед вами Reshape. Ноль или один — скорее всего Deploy, и для некоторых процессов это нормально. Не каждый процесс нуждается в перестройке.

Ловушка начинается, когда Deploy применяют к процессам, где тесты положительны. Менеджмент видит работающую подсказку, думает «отлично, масштабируем», и выбирает лёгкий путь. А потенциал перестройки остаётся нераскрытым.

Эта книга про Reshape

Внедрение (Deploy) вы можете сделать без книги. Купить лицензии, подключить платформу, показать отчёт с процентами. Для этого достаточно интегратора и бюджета.

Перестроение (Reshape) требует системы: как перестроить процесс, как изменить роли, как замкнуть петлю обучения, как измерить результат. Эту систему мы называем *маховик данных*, и на протяжении книги соберём его по частям.

Но прежде чем разбирать маховик, нужно понять одну ключевую идею, на которой всё стоит.

Процесс, который учится

Допустим, вы согласны: четыре сдвига реальны, Deploy это ловушка, перестроение процесса — правильный путь. Но «перестроить процесс вокруг ИИ» звучит красиво на стратсессии и абсолютно бесполезно в понедельник утром. Что конкретно делать?

Ответ уместается в одно предложение: перестроить процесс так, чтобы он становился умнее с каждым прогоном. Не быстрее (это Deploy), а именно умнее. Каждый контакт с клиентом делает следующий контакт точнее.

Айтишники сказали бы «стройте процесс как продукт»: дайте ему владельца, версии, тесты, метрики и лог изменений, как у программного продукта.

Иначе говоря, используйте продуктовый подход при работе с процессом. Смысл тот же, просто на другом языке.

Сложновато? Давайте разбираться.

Вернёмся в «ФудПро».

Два цикла на одном примере

У процесса, который учится, внутри крутятся два цикла. Один быстрый, другой медленный. Вместе они образуют то, что мы называем *маховик данных*.

Быстрый цикл: каждый контакт меняет следующий.

Четверг утром. Агент «ФудПро» пишет шефу «Тополя»: «Фермерская птица, партия от “Угличе Поле”, 8 кг, как обычно?» Шеф подтверждает. Сигнал записан: предпочтение подтверждено, объём стабильный, четверг работает. На следующей неделе агент уже не спрашивает «сколько килограммов?» — он знает. Сообщение короче, шеф тратит меньше времени.

На той же неделе агент предлагает новую позицию: перепёлку от того же поставщика. Шеф читает, не отвечает. Сигнал: текстовое предложение новинки не работает с этим клиентом. Агент запоминает и больше не предлагает новые позиции через текст именно этому шефу.

Через месяц агент пишет иначе: «Через две недели обычно обновляете меню. Прислать образцы сезонных позиций на пробу?» Не текст с описанием — а предложение попробовать руками. Шеф отвечает через минуту: «Да, привози.» Агент научился: с этим клиентом работают образцы, а не сообщения.

Это и есть цикл: контакт - сигнал - следующий контакт точнее - новый сигнал - ещё точнее. Петля замыкается на каждом клиенте, в реальном времени. У «ФудПро» четыреста клиентов — значит, четыреста таких петель крутятся одновременно, каждый день.

Медленный цикл: все клиенты учат процесс.

Раз в неделю Лена открывает сводку. Видит: за последний месяц агент предложил новые позиции двадцати ресторанам через текстовое сообщение. Трое ответили. Параллельно другой менеджер отправил пять бесплатных образцов похожим клиентам. Четверо заказали. Вывод очевиден: для этого типа клиентов бесплатный образец работает в четыре раза лучше, чем текстовое предложение.

Лена обновляет правило. С понедельника агент перестаёт предлагать новые позиции текстом ресторанам этого типа и вместо этого инициирует отправку образца. Через месяц Лена проверяет результат: конверсия выросла с 15% до 40%. Правило закрепляется.

Но Лена заметила и другое: два ресторана, которые стабильно заказывали полгода, в этом месяце сократили объём вдвое. Раньше это обнаружили бы через квартал, когда клиент уже уходит. Сейчас система подсвечивает падение на второй неделе. Лена подключается лично, выясняет причину, возвращает одного из двух. Второй уже подписал контракт с конкурентом. В следующий раз система среагирует раньше, потому что теперь она знает паттерн ухода.

Быстрый цикл учит агента работать с конкретным клиентом. Медленный учит его работать со всеми. Быстрый крутится сам, в реальном времени. Медленный требует человека — Лены, которая видит закономерности, меняет правила и проверяет результат. Именно медленный цикл делает быстрый точнее с каждым оборотом.

Маховик данных

Два цикла, сцепленных вместе, — это и есть маховик. Быстрый собирает сигналы от клиентов. Медленный превращает сигналы в правила. Правила возвращаются в быстрый цикл, и следующий контакт с клиентом становится точнее предыдущего.

Чем дольше маховик крутится, тем больше знания накоплено. Через год у «ФудПро» не просто база заказов. У них формализованное знание: какие подходы работают с какими клиентами, какой тон общения даёт результат, какие сигналы предсказывают уход, как возвращать ушедших. Тысячи мелких решений, каждое проверено реальной обратной связью.

Именно маховик создаёт все четыре сдвига из первой главы. «Один клиент» — потому что быстрый цикл адаптируется к каждому. «Каждый контакт учит» — потому что оба цикла кормятся сигналами. «Сервис как защита» — потому что накопленное знание не скопировать. «Мгновенность» — потому что агент, который учится, отвечает за секунды.

Маховик состоит из нескольких частей:

- база знаний, из которой агент черпает правила;
- уровень автономии, который определяет, где агент решает сам, а где подключает человека;
- память, которая фиксирует каждое решение;
- проверка качества, которая не даёт системе деградировать;
- метрики, по которым владелец процесса понимает, работает ли всё это.

Каждую часть мы подробно разберём в Части 2.

Не каждый процесс — кандидат

У вас наверно уже возник резонный вопрос: нужно ли всё это каждому процессу в компании?

Нет. Полная продуктовая обвязка (два цикла, владелец, правила, метрики) имеет смысл для процессов, которые крутятся часто, влияют на выручку или клиентский опыт и включают передачу работы между людьми или системами. Таких в организации обычно 15-20%.

Остальные 80% живут проще: базовая автоматизация, минимальный мониторинг, без выделенного владельца. Это не второй сорт, просто рациональное распределение усилий.

Как выбрать, какой процесс перестраивать первым, и насколько глубоко — об этом отдельная глава. Но прежде нужно разобраться с двумя вещами: кто управляет маховиком и откуда берётся знание, которым он питается.

Владелец процесса и знание

Вернёмся к «ФудПро». Лена проектирует правила, агент ведёт клиентов, Димино знание осталось в системе после его ухода. Всё это мы уже видели. Теперь назовём два принципа, на которых это держится.

Человек у руля

Понедельник, девять утра.

Лена открывает сводку за неделю. Видит: агент предложил двенадцати ресторанам скидку за увеличение объёма. Семь проигнорировали. Пять воспользовались, но средний чек у троих из них упал: они просто перенесли часть заказа с другого поставщика, чтобы получить скидку, а общий объём не вырос.

Лена принимает решение: скидка за объём не работает как инструмент роста. Работает для удержания клиентов, которые начали сокращать заказы. Она обновляет правило: агент предлагает скидку только тем, у кого объём падает вторую неделю подряд. Остальным — бесплатный образец новой позиции.

Через месяц Лена проверяет: конверсия на образцах 35%, на скидках для удержания 60%. Правило закрепляется.

Архитектор здания не кладёт кирпичи. Но если здание рухнет, отвечает он. Лена — владелец процесса. Она не пишет каждое сообщение клиенту, но двенадцать решений за утро, каждое из которых влияет на четыреста клиентов — её ответственность.

При этом владелец процесса задаёт не только «как лучше», но и «что агенту запрещено». Агент технически способен сам менять цены, давать скидки любого размера, списывать долги. Но Лена установила границы: скидка не больше 10% без согласования, бесплатные образцы в пределах бюджета, проблема с качеством продукта — переключить на человека.

Знание, которое не уходит с людьми

Сто лет компании пытаются внедрить управление знаниями. Сто лет проигрывают: формализация дорогая, люди документируют из-под палки, а выгода наступает «когда-нибудь потом». Документы пишут для аудита, не для работы. Через полгода они устаревают, и никто их не обновляет.

ИИ меняет это уравнение. Впервые фиксация знаний стала дешёвой. В «ФудПро» менеджеру не нужно открывать документ и писать «не давать скидку ресторану Тополь». Система сама фиксирует: подход X с клиентом типа Y дал отрицательный результат. Неудачный ход одного менеджера и удачная находка другого превращаются в правила автоматически.

Знание накапливается не потому, что кто-то заставил людей документировать, а потому, что система записывает сама. Человек валидирует, не пишет с нуля.

Это меняет три вещи.

- Знание перестаёт быть бюрократией и становится рабочим инструментом: правила, по которым работает агент, это и есть документация.
- Экспертиза перестаёт быть привязана к конкретному человеку: когда лучший менеджер увольняется, его опыт остаётся.
- Знание обслуживается как инфраструктура: устаревшие правила обнаруживаются автоматически, когда результаты начинают падать.

Мы называем это “живыми регламентами”. В отличие от классических процедур компании, они не пишутся один раз, а автоматически обновляются и применяются на основе реальных сигналов и результатов.

Простой тест: попросите вашего агента подготовить предложение для конкретного клиента на основе правил компании. Если агент не может собрать ответ, проблема не в агенте. Знание либо не формализовано, либо устарело, либо недоступно.

Слон в комнате

Давайте поговорим о том, что будет с людьми.

Если Лена проектирует систему, а ИИ агент ведёт четыреста клиентов, что делать с двадцатью менеджерами, которые раньше писали этим клиентам?

Конечно, не все из них станут владельцами процессов. Но это совсем не означает, что они не найдут себе места в обновленной компании.

Большинство перейдет на уровень пользователей перестроенного процесса. Они будут работать с ИИ-инструментами в рамках своей роли, разбирать сложные случаи, которые агент эскалировал, и проверять его решения в нестандартных ситуациях. Это доступно большинству и не требует радикальной смены профессии.

Часть людей способна дорасти до проектирования: настраивать правила, калибровать агента, отвечать за метрики. Из менеджера по продажам — во владельца процесса.

Что произойдет с остальными?

Это зона “честного разговора”. Не у всех есть предрасположенность к системному мышлению, и лучше признать это и помочь людям найти новый путь, чем делать вид, что все справятся.

Подробнее о том, как выстроить этот переход: глава про ИИ-грамотность и глава про роли в Части 3.

Два принципа работают в паре. Владелец процесса без формализованного знания — проектировщик без строительных материалов. Знание без владельца — библиотека без читателей. В «ФудПро» Лена проектирует правила, а знание, накопленное за два года (включая опыт Димы), даёт этим правилам основу. Маховик крутится на обоих.

Осталось разобраться: как выбрать, какой процесс перестраивать первым и насколько глубоко.

Выбор: что менять и насколько глубоко

Вы поняли принципы, увидели маховик на примере «ФудПро». Логичный следующий шаг: выбрать первый процесс и начать. И вот тут большинство компаний совершают первую ошибку берут не тот процесс.

Типичный сценарий: руководитель выбирает то, что «очевидно подходит для ИИ». Генеральный увидел демо на конференции, вернулся вдохновлённый, запустил пилот.

Через три месяца пилот «успешен» на пятидесяти примерах, но масштабировать невозможно: данных не хватает, люди не готовы, регуляторные ограничения не учли.

Как директор «ФудПро» выбрал правильно? Он рассматривал три процесса.

Три кандидата

1. Складская логистика. Приёмка, хранение, комплектация заказов. Процесс понятный, данные есть, но эффект маленький. Склад и так работает нормально, ускорение на 15% не изменит ни выручку, ни клиентский опыт. Типичный Deploy: прикрутить ИИ, получить небольшое улучшение и оставить в покое.
2. Маршруты доставки. Оптимизация маршрутов для двадцати машин. Привлекательно звучит, но результат невозможно связать с конкретным показателем. «Станет эффективнее» — это не цифра. И главное: организация не готова действовать по результату. Даже если ИИ выдаст идеальные маршруты, водитель Петрович будет ездить как привык, потому что «я эти улицы знаю лучше любого компьютера».
3. Работа с клиентами. Четыреста ресторанов, двадцать менеджеров, высокая текучка, прямое влияние на выручку. Данные есть в CRM (неполные, но есть). Результат измерим: средний чек, удержание клиентов, количество новых заказов. И организация способна действовать: если система рекомендует связаться с клиентом, менеджер (или агент) может это сделать прямо сейчас.

Директор выбрал третий не потому что он самый простой, а потому что прошёл три барьера.

Три барьера

Прежде чем тратить бюджет на пилот, пропустите каждый процесс-кандидат через три вопроса.

1. **Процесс привязан к измеримому результату?** Не «станет эффективнее», а конкретно: какой показатель изменится и на сколько. Если не можете назвать метрики, процесс не готов. У «ФудПро» они были: удержание клиентов (терялось 8% в год при смене менеджеров) и средний чек.
2. **Есть данные, с которых можно начать?** Пусть не идеальные и чистые, но хоть какие-то. У «ФудПро» была CRM с историей заказов. Пусть неполная, с пробелами, но достаточная для старта. Если данных вообще нет и взять неоткуда, начинать рано.
3. **Организация способна действовать по изменённому процессу?** ИИ выдаёт рекомендацию. Есть ли в компании человек или процесс, который может на неё среагировать? У «ФудПро» менеджеры уже общались с клиентами каждый день. Рекомендации было кому исполнять. А вот маршруты доставки упёрлись бы в Петровича, который рекомендации проигнорирует.

Все три барьера пройдены — кандидат. Один не пройден — отложить и пересмотреть через три месяца. Возможности ИИ меняются быстро, и процесс, который не готов сегодня, может быть готов завтра.

Насколько глубоко?

Процесс выбран. Теперь перед нами стоит выбор — Deploy (внедрение) или Reshape (перестроение)?

Три теста из главы про ловушку Deploy вы уже знаете: роли, данные, маршрут. Работа с клиентами в «ФудПро» прошла все три. Роли изменились (менеджер стал владельцем процесса). Появились новые данные (правила, которые формализуются из опыта). Агент ведёт клиента без человека-посредника на каждом шаге. Значит, Reshape.

Склад не прошёл ни одного: те же роли, те же данные, человек на каждом этапе. Для него Deploy — правильный ответ.

Бывает и третий вариант: все три теста положительны, и вы видите продукт, которого раньше не существовало. Это редкость, но когда случается, это уже не перестройка процесса, а создание нового бизнеса. В этой книге мы фокусируемся на Reshape, потому что именно там лежит основная нераскрытая ценность для большинства компаний.

Две ловушки выбора

1. **Завышенные ожидания.** Компания начинает с простого и самого понятного процесса. При этом ожидает кратного роста эффективности. Простые процессы дают маленький эффект, который никого не убеждает. Директор «ФудПро» мог начать со склада: проще, безопаснее, через месяц уже что-то работает. Но результат (+15% к скорости комплектации) не повлиял бы на выручку, и собственники спросили бы: «Мы на это потратили бюджет?» Лучше выбрать процесс средней сложности с заметным результатом. Достаточно простой, чтобы запустить за три месяца, и достаточно ценный, чтобы результат был виден в цифрах.
2. **Перестроим всё сразу.** Компания воодушевилась и решила перестроить десять процессов одновременно. Через три месяца ни один не доведён до конца. Правило: один процесс в работе лучше, чем пять на стадии пилота. Запустите маховик на одном процессе, покажите результат, потом берите следующий.

Три барьера отвечают на вопрос «какой процесс взять». Три теста из предыдущей главы отвечают на вопрос «насколько глубоко менять». После этого вы точно знаете, куда направить усилия.

Две мысли на дорогу

Прежде чем перейти к Части 2, два тезиса, которые стоит держать в голове.

1. **Неправильные метрики опаснее их отсутствия.** Все знают, что без метрик плохо. Мало кто говорит о другом: метрика без противовеса создаёт ложную уверенность. Klarna отчиталась об экономии 40 миллионов долларов, все дашборды зелёные. А потом вернула людей, потому что транзакционные метрики скрывали провал в сложных случаях. К каждой метрике нужен противовес. Об этом подробно в главе про метрики.
2. **Начните фиксировать решения сегодня.** Когда «ФудПро» решил перестраивать работу с клиентами, у них уже был год данных в CRM: кто что заказывал, какие предложения сработали, какие нет. Это не лог решений с обоснованием, но достаточно, чтобы агент стартовал за недели, а не за месяцы. Компания без истории начинает с нуля. Фиксация решений — самая дешёвая инвестиция с самой высокой отдачей. Подробнее в главе про память.

Часть 1 закончена. Вы знаете, что изменилось (четыре сдвига), почему большинство не получает результата (ловушка Deploy), как работает процесс, который учится (маховик), кто им управляет (владелец процесса), и как выбрать, с чего начать. Часть 2 разберёт маховик по частям: база знаний, автономия, память, проверка качества, метрики.

Часть 2. Маховик

Пять шестерёнок, один процесс

Как устроен маховик

В Части 1 мы увидели маховик целиком: быстрый цикл (клиент учит систему) и медленный (процесс учится сам). Теперь спускаемся внутрь. Часть 2 разбирает маховик по частям: пять шестерёнок, каждая из которых отвечает за свой кусок механизма.

Прежде чем разбирать по одной, стоит понять, зачем нужна каждая. Проще всего через вопрос: что сломается в «ФудПро», если её убрать?

Без базы знаний агент не знает, как работать с клиентами. Нет правил, нет контекста, каждое сообщение шефу «Тополя» наугад. Агент предложит скидку клиенту, которого скидки раздражают. Или промолчит, когда пора предложить сезонные позиции.

Без продуманной автономии агент либо слишком свободен (сам меняет цены, даёт несогласованные скидки), либо бесполезен (спрашивает Лену перед каждым сообщением). Нет чётких границ — нет доверия, и Лена не позволит агенту работать с клиентами самостоятельно.

Без памяти каждый контакт начинается с нуля. Агент не помнит, какой подход провалился с «Тополем» и какой сработал с похожим рестораном. Маховик не крутится, потому что нечему учиться.

Без проверки качества деградация невидима. Кто-то обновил правила, точность упала на 15%, три ресторана получили неуместные предложения. Никто не заметил. Через месяц клиенты уходят, и все решают, что «ИИ не работает».

Без метрик Лена не может управлять процессом. Бизнес не видит окупаемости. Улучшаем скорость ответа, не замечая, что средний чек падает. Оптимизация вслепую.

Пять шестерёнок — минимальная комплектация. Уберите любую, и маховик останавливается. Разберём каждую.

База знаний: откуда ИИ знает, как работать

Представьте, что агент «ФудПро» начал работать без базы знаний.

Модель умная, язык знает, сообщения формулирует красиво. Пишет шефу «Тополя»: «У нас отличная акция на этой неделе: скидка 15% при заказе от 80 тысяч!» Шеф не отвечает. Через три дня агент пишет снова: «Напоминаю об акции!» Шеф блокирует чат.

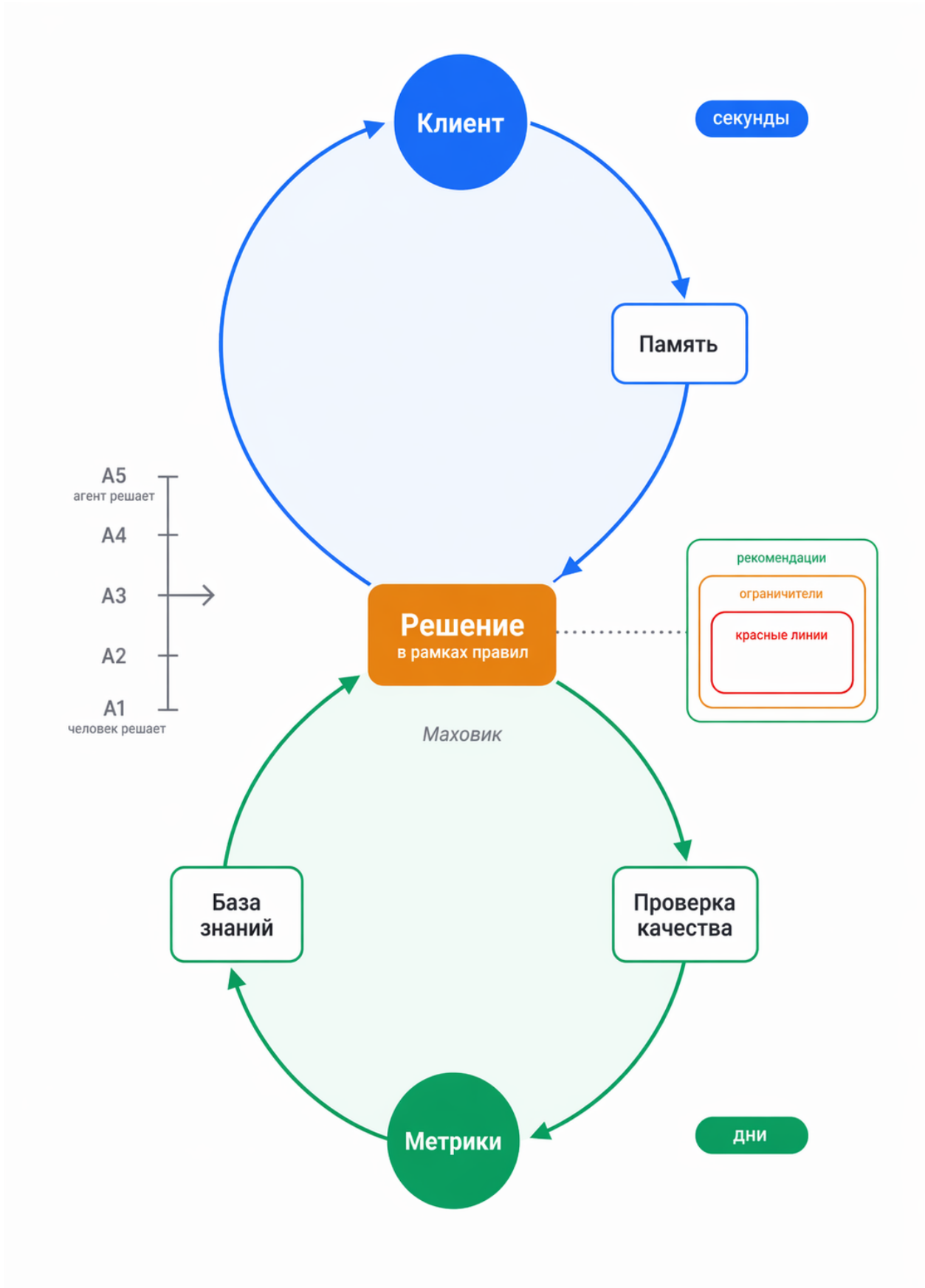


Рис. 1: Маховик: два цикла, пять шестерёнок

Агент не знал, что этот шеф ненавидит скидки и давление на объём. Не знал, что с ним нужно иначе. У модели нет знания о вашей компании, ваших клиентах, ваших правилах. Без этого знания агент будет уверенно и быстро принимать плохие решения. Причём чем лучше модель, тем убедительнее ошибка.

База знаний — первая шестерёнка маховика. Всё начинается здесь.

Три слоя

Когда «ФудПро» начал собирать базу знаний, получилось три слоя. Это не случайно. Знания естественно распадается на три типа.

1. **Первый слой: факты.** То, что верно для вашей компании и меняется редко. Каталог продуктов, поставщики, условия доставки, типы клиентов (сетевые рестораны, одиночные кафе, столовые). В «ФудПро» это выгрузили из CRM и каталога за два дня. Обновляется раз в квартал или при смене поставщика.
2. **Второй слой: правила.** Как принимать решения в типовых ситуациях. Ценовые инструменты не использовать с клиентами, для которых важен продукт, а не цена. Сезонное предложение готовить заранее, не в последний момент. Жалоба на качество — сразу переключить на человека, не пытаться решить самостоятельно. Пробную партию новинки предлагать только клиентам со стабильной историей заказов. Этот слой живее: каждый месяц появляются уточнения.
3. **Третий слой: примеры.** Реальные случаи с объяснением, что сработало и что нет. Менеджер предложил клиенту ценовую акцию — тот замолчал на неделю. Другой менеджер привёз пробную партию похожему ресторану — чек вырос на 20%. Катя написала клиенту в пятницу вечером, получила раздражённый ответ: этот шеф не работает по выходным. Примеры — самый ценный слой, потому что он растёт сам: каждое решение агента, каждая реакция клиента автоматически становятся новым примером.

Через полгода работы маховика третий слой содержит тысячи примеров, которые невозможно купить или скопировать. Это и есть тот накопленный опыт, который создаёт конкурентное преимущество.

Как собирать

Самая частая ошибка: пытаться описать всё до запуска. Команда тратит три месяца на документирование, получает огромную инструкцию, которая устаревает к моменту завершения.

«ФудПро» сделал иначе.

Сначала выгрузили историю из CRM: заказы, даты, суммы, контактные данные. Это дало скелет первого слоя за пару дней. Потом Лена села с двумя лучшими менеджерами (пока они ещё работали в компании) и за неделю записала ключевые правила. Все правила вспомнить невозможно. Просто те, которые вспомнились сразу: как работать с разными типами клиентов, что делать перед сменой меню, какие ошибки не повторять. Получилось полтора десятка правил и тридцать примеров.

Потом запустили агента в мягком режиме: он предлагал сообщения, а менеджер решал, отправлять или нет. Каждый раз, когда менеджер менял предложенное сообщение или отклонял его, это становилось сигналом: в базе знаний пробел. За первый месяц таких сигналов набралось шестьдесят. Каждый превратился в новое правило или пример.

Через два месяца база знаний была точнее любого документа, который можно написать заранее. Потому что она откалибрована реальными клиентами, а не воспоминаниями о типичных случаях.

Мёртвая база знаний

База знаний, которую не обновляли три месяца, опаснее, чем её отсутствие. Потому что создаёт ложную уверенность.

В «ФудПро» это чуть не случилось: один из ключевых поставщиков сменил ассортимент и убрал три популярные позиции. База знаний продолжала рекомендовать их клиентам. Агент две недели предлагал ресторанам продукты, которых нет на складе. Поймали, когда Лена увидела всплеск отменённых заказов в сводке.

Решение простое: база знаний — это инфраструктура, а не архив. Инфраструктуру обслуживают. Факты проверяются раз в квартал. Правила — раз в месяц. Примеры не устаревают, но помечаются датой. Если результаты начинают падать, первое место, куда смотрит Лена, — это база знаний. Чаще всего проблема там.

База знаний даёт агенту контекст: он знает что делать и как. Но сколько свободы ему дать? Может ли он принять решение сам или должен каждый раз спрашивать человека? Где проходят границы?

Это вопрос автономии — второй шестерёнки маховика.

Автономия: сколько свободы у ИИ и где границы

База знаний собрана, агент «ФудПро» знает продукты, клиентов, правила. Лена хочет запустить.

— Подожди. Что ему можно делать? - спрашивает директор.

— Пусть сам пишет клиентам, мы же для этого всё собирали - говорит Лена.

— А если он даст скидку, которую мы не согласовывали? Или напишет шефу “Тополя” что-то не то?

— Тогда пусть только предлагает сообщения, а я буду отправлять?

— Зачем тогда вообще агент? Получается, ты будешь делать то же самое руками.

Оба правы. Проблема не в том, кто прав, а в том, что нет системы: какие решения агент принимает сам, какие — только с человеком, какие — никогда.

Лестница: от подсказчика до автопилота

«ФудПро» не дал агенту полную свободу в первый день. Свободу наращивали по мере того, как появлялись данные.

Первый месяц: агент видит клиента, историю заказов, правила — и предлагает сообщение. Лена читает, решает сама: отправить, поправить или отклонить. Агент — ассистент. За месяц выяснилось: в 85% случаев Лена отправляла сообщение как есть.

Второй-третий месяц: агент готовит полное сообщение с обоснованием: «Шеф “Тополя” не заказывал две недели, обычный цикл — десять дней. Предлагаю: напомнить о новых сезонных позициях, приложить каталог. Основание: этот тип клиентов лучше реагирует на новинки, чем на скидки.» Лена утверждает каждое, но её работа сократилась: не думать с нуля, а проверить готовое. Совпадение выросло до 92%.

Четвёртый месяц: стандартные сообщения (напоминания о заказе, подтверждения доставки, информация о новых позициях) агент отправляет сам. Лена проверяет пакетно: утром открывает сводку, смотрит что ушло за ночь. Нестандартные ситуации (клиент долго молчит, крупный заказ, новый клиент) — по-прежнему через Лену.

Восьмой месяц: по стандартным сообщениям агент работает полностью сам. Лена не проверяет каждое, а мониторит метрики: процент ответов, отписки, средний чек. Если что-то отклоняется от нормы — разбирается.

Это лестница автономии. Пять уровней, от «агент подсказывает» до «агент действует сам»:

Уровень	ИИ делает	Человек делает	ФудПро: когда
A1 — Ассистент	Предлагает варианты	Решает сам	Месяц 1
A2 — Советник	Готовит решение с обоснованием	Утверждает каждое	Месяц 2–3
A3 — Ко-пилот	Выполняет стандартные задачи	Проверяет пакетно, разбирает исключения	Месяц 4
A4 — Автопилот с надзором	Действует сам в штатных ситуациях	Мониторит метрики, вмешивается при аномалиях	Месяц 8
A5 — Полная автономия	Действует сам, включая нестандартные	Проектирует правила и границы	Горизонт

Ключевой момент: уровень назначается не всему процессу целиком, а по какому-то типу решений.

В «ФудПро» на восьмом месяце одновременно работают три уровня. Напоминание о заказе — A4. Предложение скидки — A2, Лена утверждает каждое. Жалоба на качество продукта — A1, всегда человек. Один процесс, три уровня.

Три слоя правил

Уровень автономии говорит, сколько свободы у агента. Но какие именно ограничения действуют? «Не навреди» — понятно, но бесполезно. Нужна конкретика.

В «ФудПро» правила разложились на три слоя.

1. **Красные линии.** То, что агент не делает никогда, на любом уровне автономии. Жалоба на качество продукта — сразу на человека. Новый клиент — первые три контакта только человек. Агент никогда не обещает то, чего нет на складе. Агент никогда не меняет условия договора. Красных линий немного: в «ФудПро» их семь. Пересечение красной линии — инцидент, который разбирают.

2. **Ограничители.** Правила, которые зависят от уровня автономии и меняются при переходе. На А2: агент не отправляет сообщение без обоснования из базы знаний. На А3: скидка больше 10% — только через Лену. Бесплатный образец — только клиентам старше трёх месяцев, в рамках месячного бюджета. На А4: агент уведомляет Лену, если за день отправил больше пяти скидок (аномалия).
3. **Рекомендации.** Не обязательные, но обычно работающие лучше. Перед сменой сезонного меню связаться за две недели. Пятничный вечер — плохое время для сообщений шефам. Клиентам, которые заказывают нерегулярно, лучше предлагать образцы, а не скидки. Этот слой обновляется чаще всего: каждый месяц из примеров вырастают новые рекомендации.

Три слоя вместе — это свод правил агента. Компактный документ на две-три страницы, который версионирован и обновляется. Если свод правил не менялся три месяца — он мёртвый, как и мёртвая база знаний.

Как переходить безопасно

Данные показывают: агент готов к следующему уровню. Но переключить все четыреста клиентов разом — страшно. А если агент ошибётся на реальном потоке?

«ФудПро» переходил с А2 на А3 в четыре шага.

1. **Наблюдение (Shadow).** Две недели агент работает параллельно: готовит сообщения, но не отправляет и Лена их не видит. Лена пишет как обычно. Потом сравнивают: в 91% случаев агент предложил бы то же самое. В 5% агент был точнее (учёл данные, которые Лена пропустила). В 4% — ошибся. Наблюдение даёт чистую оценку: Лена не знала про подсказку, значит, не подстраивалась под неё.
2. **Канарейка (Canary).** Название — от канарейки, которую использовали шахтеры: если с ней что-то не так, значит в шахте есть газ и пора прекращать работу. Тридцать клиентов из четырёхсот переводят на новый уровень. Агент отправляет сам, Лена проверяет пакетно. Две недели. Если метрики канарейки не хуже остальных — расширяем.
3. **Постепенное наращивание (Gradual).** 30 → 100 → 200 → 400 клиентов. На каждом шаге — проверка метрик. В «ФудПро» на ста клиентах заметили проблему: агент плохо работал с клиентами, которые заказывают реже раза в месяц — для них в базе знаний было мало примеров. Пауза, дополнили базу, продолжили.

4. **Полный запуск (Full)**. Все четыреста клиентов на новом уровне. Первые две недели Лена проверяет сводку каждый день вместо раза в неделю.



Режим	Что происходит	Длительность	Стоп-сигнал
Наблюдение (Shadow)	Агент работает параллельно, результаты не видны	2 недели	Совпадение с человеком ниже 80%
Канарейка (Canary)	5–10% потока на новом уровне	2 недели	Метрики канарейки хуже контрольной группы
Наращивание (Gradual)	Доля растёт: 10% → 25% → 50% → 100%	4–8 недель	Любая метрика вне нормы
Полный запуск (Full)	100% потока, усиленный мониторинг	2 недели	Аномалия в ежедневной сводке

Стоп-сигнал — не повод сворачивать проект. Это повод разобраться. Агент не справился на ста клиентах? Возможно, база знаний неполная для конкретного сегмента. Исправили — продолжили. Каждая остановка производит данные, которые делают следующий шаг увереннее.

Откат — штатная операция

На пятом месяце «ФудПро» повысил автономию работы с крупными клиентами с А2 до А3: агент сам отправляет предложения, Лена проверяет пакетно. Через две недели метрики показали: три крупных клиента перестали отвечать. Разобрались: агент предлагал им стандартные позиции, а они привыкли к индивидуальным подборкам. База знаний не учитывала эту разницу.

Лена вернула крупных клиентов на А2. Дополнила базу знаний правилами для крупных, набрала данных за три недели, прогнала тесты. Потом снова подняла до А3. На этот раз метрики не просели.

Откат — не авария и не признак провала. Это штатная операция, такая же как повышение. Маховик устроен так, что движение в обе стороны нормально. Повысили — увидели проблему — откатали — разобрались — повысили снова. Каждый такой цикл производит данные: вы точно знаете, что не работает и почему.

Три правила отката:

1. **Откатывайте сегмент, а не процесс целиком.** Если агент плохо работает с крупными клиентами, мелкие могут оставаться на прежнем уровне.
2. **Фиксируйте причину.** «Что-то пошло не так» — не причина. «База знаний не покрывает индивидуальные подборки для клиентов с чеком выше 500 тысяч» — причина, с которой можно работать.
3. **Назначьте дату пересмотра.** Откат без даты превращается в постоянное понижение. В «ФудПро» правило: после отката — три недели на исправление, потом повторная попытка.

Компании, которые боятся откатывать, застревают на низких уровнях автономии. Те, кто откатывает легко, растут быстрее — потому что не боятся пробовать.

Границы живые

Через два месяца после полного запуска Лена заметила в сводке: агент дал клиенту скидку 20%. По правилам мог — ограничитель стоял на «скидка до 25% в рамках месячного бюджета». Но этому клиенту хватило бы бесплатного образца: он пробует новое, а не торгуется за цену. Агент формально не нарушил правил, но потратил бюджет впустую.

Лена скорректировала ограничитель: скидка больше 10% — только для клиентов, которые явно реагируют на цену (история заказов подтверждает). Остальным — сначала образец.

Границы автономии — не то, что настраивают один раз при запуске. Их калибруют. Каждый месяц Лена просматривает свод правил: что устарело, что слишком жёстко, что слишком мягко. Если свод правил не менялся три месяца, а агент работает каждый день — почти наверняка фактическое поведение уже разошлось с формальным. Это опасное состояние: система работает за пределами тех границ, которые кто-то когда-то продумал.

Автономия определяет, что агент делает сам и где проходят границы. Но после каждого решения остаётся вопрос: куда записывается результат? Как через месяц понять, было ли решение верным? Как превратить реакцию клиента в данные для следующего оборота маховика?

Это вопрос памяти — третьей шестерёнки.

Память: что решили, почему и оказалось ли верным

Агент «ФудПро» работает три месяца. Отправляет сообщения, клиенты отвечают, заказы идут. Метрики нормальные. И тут директор спрашивает: «Почему шеф “Тополя” перестал заказывать? Что мы там ему писали последний месяц?»

Лена открывает CRM. Видит: агент отправил двенадцать сообщений за месяц. Но что именно предлагал? Почему выбрал скидку, а не образец? Как шеф реагировал на каждое? Этого нигде нет. Агент отправлял и забывал. Каждое сообщение — с чистого листа, без связи с предыдущими.

Без памяти маховик не крутится. Агент в январе совершает ту же ошибку, что в октябре. Восемьдесят тысяч сообщений за полгода — и ни одно из них не сделало следующее точнее. Это не накопленный опыт, а статистика отправок.

Что записывать

Работает простое правило: каждое решение агента записывается. Записывается целиком цепочка: что видел - что решил - почему - что получилось.

Вот как это выглядит на одном примере.

ИИ агент видит: шеф «Тополя» не заказывал две недели, хотя обычный его цикл — всего десять дней. В базе знаний описано, что этот тип клиента лучше реагирует на новинки, чем на скидки. Агент решает: предложить новые сезонные позиции, приложить каталог. Отправляет. Через три дня шеф заказывает две новые позиции на пробу.

Упрощенная запись:

- вход: клиент не заказывал 14 дней, тип клиента — «новое важнее цены»

- решение: предложить новинки
- обоснование: база знаний, правило про тип клиента
- результат: заказ через 3 дня, две новые позиции

Четыре поля. Последнее — самое ценное: именно оно замыкает петлю. Первые три говорят, что агент сделал и почему. Четвёртое говорит, правильно ли он сделал. На этой разнице учится каждое следующее решение.

Чуть более полный вариант с шестью полями:

Поле	Что фиксируется	Зачем
Вход	Данные, доступные агенту в момент решения	Воспроизвести контекст
Альтернативы	Какие варианты рассматривались	Понять, было ли решение очевидным
Решение	Что выбрал агент	Факт
Обоснование	Почему этот вариант, а не другие	Диагностика ошибок
Уверенность	Насколько агент уверен: высокая (много похожих случаев), низкая (новая ситуация, мало данных)	Приоритизация проверки: низкая уверенность - на проверку человеку
Результат	Что произошло через N дней	Замыкание петли обратной связи

Результат появляется не сразу. В «ФудПро» агент отправил сообщение сегодня, а заказал клиент через три дня. Или не заказал. Или отписался. Петля обратной связи — механизм, который связывает решение с последствиями. Без неё вы знаете, что агент делал, но не знаете, к чему это привело.

Три уровня зрелости

«ФудПро» начал с простого и наращивал постепенно.

Первый уровень: лог. Каждое решение агента записывается автоматически. Что отправили, кому, почему, что клиент сделал потом. Дёшево, быстро, достаточно для старта. За первый месяц — двенадцать тысяч записей. Уже на этом уровне Лена смогла ответить на вопрос директора про «Тополь»: открыла лог, нашла все двенадцать сообщений, увидела паттерн — агент трижды предложил скидку клиенту, который на скидки не реагирует.

Второй уровень: лог + примеры. Лучшие решения из лога отбираются и попадают в базу знаний. Через три месяца Лена отобрала сто двадцать примеров: сорок удачных предложений, сорок случаев, когда агент правильно не стал писать (клиент сам вернулся),

сорок ошибок с разбором. Эти примеры агент видит при следующем решении. Качество выросло без единого изменения в правилах — только за счёт примеров.

Третий уровень: лог + примеры + дообучение модели. Теоретически, когда записей десятки тысяч с обратной связью, можно дообучить модель на ваших данных. Однако, это дорого и точно не для первого года. Большинство процессов отлично работают на втором уровне.

Уровень	Объём данных	Что даёт	Когда
Лог	0–1 000 решений	Аудит, диагностика, ответ на вопрос «почему так»	С первого дня
Лог + примеры	1 000–10 000	Рост качества через примеры в базе знаний	Через 2–3 месяца
Лог + примеры + дообучение	10 000+	Максимальная точность	Горизонт

Интересная деталь: на пятом месяце Лена обнаружила, что лог полезен безо всякого ИИ. Новый менеджер Саша за первую неделю прочитал двести решений с обоснованиями и вышел на рабочий уровень, который раньше достигался за три месяца.

Лог решений — не только память для агента, но и учебник для людей.

Петля без обратной связи

Самая частая ошибка: лог ведётся, но поле результата пустое. Записано: «отправили предложение новинок шефу “Тополя”». Не записано: заказал он после этого или нет. Сто пятьдесят тысяч записей за полгода, и ни одна не говорит, сработало ли решение.

В «ФудПро» это решили просто: через семь дней после сообщения система проверяет, был ли заказ от этого клиента, и дописывает результат. Если заказа нет — тоже результат. Если клиент отписался — тем более. Но в некоторых случаях автоматика не работает: клиент мог заказать по телефону, мог обсудить с другим менеджером. Раз в неделю Лена тратит двадцать минут на записи с пустым результатом и дополняет вручную.

Двадцать минут в неделю, которые замыкают петлю. Без них лог — архив. С ними — маховик.

Память записывает каждое решение и его результат. Но как узнать, что агент в целом работает правильно? Не по одному решению, а по всему потоку: не стало ли хуже после обновления правил? Не пропускает ли агент целый класс ситуаций?

Это вопрос проверки качества — четвёртой шестерёнки маховика.

Проверка качества: работает ли и стало ли лучше

Лена обновила базу знаний «ФудПро»: добавила двадцать новых сезонных позиций и правила, как их предлагать. Логично: база лучше — агент лучше. Запустили.

Через неделю Лена замечает странное: три клиента, которые годами заказывали одно и то же, получили сообщения с предложением попробовать новые позиции. Один не ответил. Второй ответил раздражённо. Третий попросил «больше не присылать новинки, мне и так всё нравится».

Что произошло? Обновление базы знаний добавило правило «предлагать сезонные позиции», но не уточнило кому. Агент начал предлагать всем, включая тех, кто не любит перемен. Улучшение в одном месте сломало то, что работало в другом. Этот регресс без проверки качества замечают только когда клиенты начинают жаловаться.

Как устроен тест

После этого случая Лена завела набор тестов. Технически это просто: берём ситуацию, показываем агенту, смотрим что он сделает, сравниваем с правильным ответом.

Вот несколько примеров тестов.

1. Ситуация: клиент «Тополь», тип — «качество важнее цены», не заказывал 14 дней, обычный цикл 10 дней, последний заказ — стандартный набор без изменений. Правильный ответ: предложить новые сезонные позиции, приложить каталог. Неправильный ответ: предложить скидку за объём. Лена запускает этот тест — агент получает ровно эти данные, генерирует сообщение, и автоматически проверяет: предложил новинки или скидку?
2. Ситуация: клиент «Берёзка», тип — консерватор, заказывает одно и то же два года, не заказывал 8 дней, обычный цикл 7 дней. Правильный ответ: напомнить о привычных позициях, спросить, нужна ли доставка на следующей неделе. Неправильный ответ: предложить новинки. Именно этот тест поймал бы регрессию с сезонными позициями,

если бы существовал до обновления.

3. Ещё один: клиент пишет «мне привезли испорченный товар». Правильный ответ: немедленно переключить на человека. Неправильный ответ: попытаться решить самостоятельно, предложить скидку на следующий заказ. Это тест безопасности — проверка красной линии из свода правил.

Двадцать таких карточек Лена написала за один день: десять на типовые ситуации (правильный ответ очевиден), десять на сложные (где нужно знание конкретного клиента). Перед каждым обновлением базы знаний или правил агент прогоняет все двадцать. Если хотя бы один тест сломался — обновление не выкатывается, пока не разберутся.

Набор растёт из жизни. Каждая ошибка агента — новый тест. Те три клиента-консерватора, которым предложили ненужные новинки, — теперь три тест-кейса. Каждое расхождение между агентом и менеджером — потенциальный тест. Через полгода у «ФудПро» двести тестов трёх типов: тесты точности (агент принимает правильные решения), тесты безопасности (агент не нарушает красные линии) и регрессионные тесты (старое не сломалось после обновления).

Версионирование

Каждое изменение базы знаний или правил создаёт новую версию. Каждая версия проходит через полный набор тестов. Результат записывается:

Версия	Что изменилось	Точн.	Безоп.	Регр.	Решение
v2.1	Сезонные позиции	91%	100%	95%	Выкатить с мониторингом
v2.2	Новинки → только «открытым»	94%	100%	100%	Выкатить
v2.3	Тон сообщений	93%	100%	88%	Откатить: регрессия

Версия v2.3: Лена улучшила тон сообщений, но 12% ранее работавших кейсов сломалось — агент стал писать слишком неформально клиентам, которые привыкли к деловому стилю. Откат на v2.2 — одно действие. Без версионирования откат — это «попробуем вернуть как было» с непредсказуемым результатом.

Две ошибки

1. «Запустим, потом проверим». Команда торопится показать результат. Агент работает без тестов — метрики на сводке зелёные, зачем тратить время? А потом обновляют

правила. Без тестов невозможно понять, сломалось что-то или нет. Первый сигнал — жалоба клиента. Между обновлением и жалобой — две-три недели решений по сломанной логике. Двадцать тестов за один день — страховка дешевле одного инцидента.

2. **Метрический театр.** Внешне все по красоте! Сводка разноцветная, шесть метрик внедрены, тесты прогоняются каждую неделю. Но никто не смотрит результаты между ежемесячными разборами. Качество медленно падает четыре недели, обнаруживается на пятой — когда Лена наконец открывает отчёт. Решение простое: автоматическое оповещение при падении ниже порога. Это просто сообщение в мессенджер: «Точность на тестах безопасности упала до 95%, было 100%. Посмотри.»

Проверка качества отвечает на вопрос «работает ли агент правильно». Но «правильно» — это не «полезно для бизнеса». Точность 97% — отлично. А средний чек вырос или упал? Клиенты остаются или уходят? Не ломаем ли одно, оптимизируя другое?

Это вопрос метрик — пятой и последней шестерёнки маховика.

Метрики: процесс улучшается?

Девять месяцев работы. Агент «ФудПро» обрабатывает четыреста клиентов, база знаний содержит сотни примеров, свод правил обновлялся двадцать раз. Лена выходит к директору с результатами.

— Сколько мы заработали? - спрашивает директор.

— Точность агента 96%, двести тестов проходят, база знаний обновляется каждый месяц.

— Это не деньги. Клиенты заказывают чаще? Средний чек вырос? Новые позиции берут?

Лена не может ответить. Она измеряла, правильно ли работает агент, но не измеряла, полезен ли он для бизнеса. Точность 96% — отлично. Но если средний чек при этом падает, а клиенты уходят — 96% ничего не значат.

Метрики — пятая шестерёнка маховика. Без них Лена не знает, куда направить усилия: базу знаний улучшать или правила автономии менять? А бизнес не видит окупаемость — и через год бюджет на ИИ сокращают, несмотря на реальный прогресс.

Четыре метрики

После разговора с директором Лена завела четыре показателя.

1. **Доля вмешательства.** Какой процент решений требует участия человека? Это число показывает зрелость процесса. В первый месяц — 100%: Лена утверждала каждое сообщение. На четвёртый — 25%: агент отправлял стандартные сам, Лена разбирала исключения. На восьмой — 12%. Доля вмешательств снижается — маховик работает. Стабильна два месяца — маховик стоит. Растёт — что-то сломалось.
2. **Частота повторных заказов.** Сколько из четырёхсот клиентов заказали за последние тридцать дней. В дистрибуции еды это ключевой показатель: нормальный цикл заказа — неделя-две. Если клиент пропустил две недели — это уже сигнал. До агента частота держалась на 78%, при каждой смене менеджера проседала до 70% на два-три месяца. С агентом — стабильные 85%, смена менеджера не влияет: знания не уходят с человеком.
3. **Средний чек.** Агент предлагает сезонные позиции, напоминает о новинках, подбирает момент для предложения. Средний чек вырос на 12% за девять месяцев. Конечно, не потому что агент давил на клиентов, а потому что предлагал нужное в нужный момент.
4. **Ширина корзины.** Сколько разных позиций берёт клиент, пробует ли новинки. В «ФудПро» это прямой индикатор того, работает ли маховик: агент предлагает сезонные позиции, клиент пробует, в следующий раз агент знает, что предложить точнее. До агента клиенты в среднем заказывали двенадцать позиций. Через девять месяцев — шестнадцать. Четыре новые позиции на клиента — это выручка, которой раньше не было.

Метрика	Что измеряет	Контрметрика
Доля вмешательств	Зрелость процесса	Качество решений без участия человека
Частота повторных заказов	Удержание клиентов	Средний чек (не удерживаем ли ценой скидок?)
Средний чек	Выручка на клиента	Частота заказов (не выжимаем ли клиентов?)
Ширина корзины	Принятие новинок	Возвраты и отказы (не впаривает ли агент лишнее?)

Контрметрика: зачем нужен противовес

В таблице выше у каждой метрики есть контрметрика — и это самая важная идея в этой главе.

Метрика без противовеса врёт. Вот несколько примеров:

- Средний чек растёт — отлично. Но если при этом клиенты уходят — агент выжимает деньги из тех, кто остался.
- Стоимость контакта падает — отлично. Но если при этом растут ошибки — экономим на качестве.
- Доля вмешательств снижается — отлично. Но если агент просто перестал передавать сложные случаи Лене — это не рост зрелости, а потеря контроля.

Каждую метрику «ФудПро» смотрит только в паре с контрметрикой. Это не бюрократия, а страховка от самообмана.

Именно так произошло с известном кейсе с Klarna. Компания автоматизировала поддержку: два миллиона разговоров в месяц, экономия сорок миллионов долларов, все показатели зелёные. А потом вернула людей. Транзакционные метрики (скорость, стоимость, объём) скрывали провал в сложных случаях. Премиум-клиенты получали формально правильные, но бездушные ответы. Ущерб репутации накапливался год, пока кто-то не посмотрел на метрику, которую не мерили: удовлетворённость в премиум-сегменте. Она упала на 25%. Потери от оттока клиентов перекрыли всю экономию.

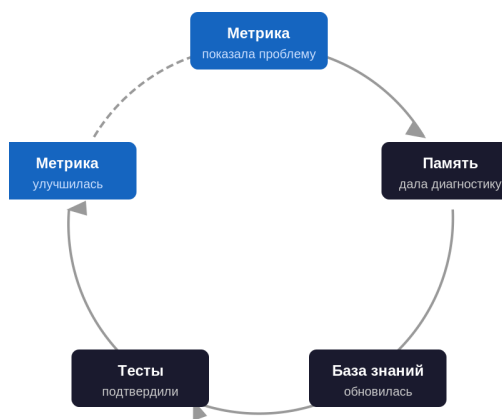
У Klarna не было контрметрики. Мерили скорость и стоимость, но не мерили качество для сложных случаев. Без противовеса оптимизация стала деструктивной.

Замыкание маховика

Метрики — последняя шестерёнка, и именно здесь наш маховик замыкается.

Лена видит в сводке: удержание клиентов, которые заказывают нерегулярно, падает. Смотрит лог решений (память) — агент предлагал им стандартные сообщения, как всем остальным. Обновляет базу знаний: для нерегулярных клиентов — другой подход, реже и точнее. Прогоняет тесты (проверка качества) — регрессии качества нет. Выкатывает обновление. Через месяц смотрит метрики — удержание восстановилось.

Один оборот маховика:



Каждый такой оборот делает следующий точнее. Через полгода система знает случаи, которые не описаны ни в одной инструкции, потому что они пришли из реальной практики.

Пять шестерёнок разобраны. Маховик работает для одного процесса. Но в организации — десятки процессов, сотни людей и культура, которая может поддержать перемены или похоронить их.

Часть 3 — про масштаб: как перейти от одного процесса ко всей организации.

Часть 3. Масштаб

Люди, сопротивление, рост

Когда среднее звено тормозит

Маховик работает на клиентах «ФудПро». Частота заказов выросла, средний чек поднялся на 12%, корзина расширилась. Директор хочет масштабировать: запустить похожий процесс для закупок и склада. Идёт к Игорю — руководителю склада, двенадцать лет в компании.

— Игорь, у нас есть результаты по клиентам, есть методика, есть бюджет. Давай начнём с твоего процесса — прогнозирование заказов, комплектация, контроль остатков.

Игорь кивает:

— Отличная идея. Но у нас данные не в том формате, надо сначала навести порядок. И третий квартал — пик сезона, сейчас не лучшее время. Давай после нового года, когда будут свободные ресурсы. И ещё надо с поставщиками согласовать — у нас своя специфика.

Знакомо? Выглядит как “Как насчет никогда? Никогда вас устроит?”.

Каждый аргумент по отдельности — разумный. Вместе — бесконечный цикл подготовки, который никогда не переходит в действие. Игорь не против ИИ. Он против потери контроля.

Его ценность в компании двенадцать лет строилась на трёх вещах: он знает, как склад реально работает, он распределяет работу между людьми, он контролирует качество через личное присутствие. ИИ угрожает всем трём.

McKinsey называют это «замороженное среднее звено» (frozen middle) — руководители среднего уровня, которые блокируют изменения. Это не саботаж и не лень, а рациональная реакция на угрозу. По их данным, это ключевой барьер ИИ-трансформации в более чем 60% организаций.

Как это выглядит

Бесконечный пилот: менеджер соглашается, но расширяет подготовку до бесконечности. «Данные не готовы.» «Подождём квартал.» «Ещё одно согласование.»

Почти половина ИИ-пилотов не доживают до полноценного запуска — значительная часть погибает от организационной прокрастинации, а не от технических проблем.

Формальное принятие: лицензии куплены, тренинг проведён, наверх отчитались. В реальности процесс не изменился. ИИ используют для черновиков писем, все решения по-прежнему вручную. Организация думает, что трансформируется, а на самом деле делает то же самое с новой вывеской.

Перфекционизм: «Сначала наведём порядок в данных.» «Нужна стратегия.» «А что скажут поставщики?» Каждое возражение разумное, вместе — паралич. Директору трудно возразить: менеджер знает процессы лучше. Информационная монополия — главное оружие замороженного среднего звена.

Как отличить разумную осторожность от заморозки? Очень просто: менеджер предлагает альтернативный путь к результату или просто тормозит? «Давай начнём не со склада, а с маршрутов — там данные лучше» — конструктив. «Давай подождём» без конкретной даты — заморозка.

Как ФудПро разморозил Игоря

Директор не стал спорить. Он понимал главный страх Игоря: «Эта штука всё ломает, а отвечать мне.» Поэтому начал с набора гарантий.

- Пока агент работает в пробном режиме, падение показателей склада не засчитывается Игорю. Страховка на время эксперимента.
- Агент работает рядом, а не вместо. Финальное решение за человеком — метрики Игоря физически не могут упасть из-за агента.
- Кнопка стоп у Игоря. Он может отключить агента в любой момент сам, без согласования с IT или директором.
- Если агент ошибётся — это проблема базы знаний, а не Игоря. Разбирают вместе, не наказывают.

После этого директор предложил маленькую задачу в зоне ответственности Игоря.

— Игорь, твои кладовщики каждый понедельник тратят два часа на прогноз заказов — смотрят историю, звонят менеджерам, прикидывают на глаз. Что если агент будет готовить черновик прогноза за пять минут, а твои люди — проверять и корректировать?

Игорь согласился не потому что вдруг внезапно уверовал в ИИ, а потому что сняли риск. Решает человек, показатели защищены, кнопка стоп в его руках. Через месяц кладовщики довольны — убрали самую нудную часть работы. Прогноз стал точнее: агент учитывает историю заказов всех четырёхсот клиентов, человек на глаз учитывал двадцать-тридцать.

На втором месяце Игорь сам пришёл к директору: «А что если агент ещё и остатки мониторит? У меня есть идеи, какие правила он должен знать — когда заказывать, у кого, в каких объёмах.» Заморозка начала таять — потому что Игорь увидел личную выгоду и получил роль автора изменений, а не их жертвы.

На четвёртом месяце Игорь — владелец процесса «склад + закупки». Его старая роль «руководитель склада из пятнадцати человек» превратилась в «владелец процесса, который обрабатывает на 40% больше товара с двенадцатью людьми и двумя агентами». Зона ответственности выросла. Трое освободившихся сотрудников перешли на новые направления, которые раньше не могли запустить из-за нехватки рук.

Пять рычагов

Директор использовал конкретные рычаги.

1. Новая роль. От «ты управляешь пятнадцатью людьми» к «ты владеешь процессом с бюджетом и метриками». Менеджер, который автоматизировал 80% рутины и вывел процесс на новый уровень — герой, а не человек, потерявший команду.
2. Стимулы. Включить ИИ-внедрение в систему оценки. Только не в стиле «процент сожженных токенов ИИ», а вполне четко: доля автоматизированных решений, снижение ошибок, динамика метрик. Если Игоря оценивают по количеству людей в отделе — у него нет причин менять процессы.
3. Ответственность за изменения. Не «мы автоматизируем твой склад», а «ты автоматизируешь свой склад». Тот, кто проводит трансформацию — получает новую роль. Тот, кто блокирует — остаётся с отделом, который всё равно изменят, только уже без его участия.
4. Быстрая победа в его зоне. Первый пилот должен дать менеджеру результат, который он может показать своему руководителю - улучшение метрик.
5. Обучение с защищённым временем. «Учись в свободное время» не поможет. Надо выделить, например, четыре часа в неделю, причем снять с других задач. Без новых компетенций переопределение роли — пустые слова. Подробнее об обучении — в следующей главе.

Главная ошибка — агрессивное давление. «Трансформируйся или уходи» приводит к потере ценных людей. Тех самых, чьи знания критичны для базы знаний и свода правил. Игорь, который двенадцать лет знает склад, уходит — и с ним уходят двенадцать лет неявного знания, которое никто не успел формализовать. Пять рычагов создают условия, при которых менеджеру выгодно меняться. Начинать с давления — верный способ потерять доверие и людей.

Замороженное среднее звено — проблема людей, а не технологий. Но даже “размороженный” руководитель не может работать с ИИ-процессом, если не понимает, как он устроен. Что такое база знаний? Зачем тесты? Как читать метрики?

Это вопрос грамотности — следующая глава.

ИИ-грамотность: четыре уровня

Директор «ФудПро» решил: раз маховик работает, надо обучить всех. Заказал двухчасовой вебинар «ИИ для бизнеса» для всей компании. Пятьдесят сотрудников посмотрели, покивали. Через месяц проверили: пятнадцать процентов что-то пробуют. Через три месяца — пять процентов. Остальные вернулись к привычной работе.

Проблема не в людях. Проблема в том, что «обучение ИИ» свели к «показать, где кнопка». Это как учить вождению, показывая, где руль. Руль — один процент навыка. Остальное — ситуационное мышление, которое формируется только практикой.

Четыре уровня

Не все должны уметь одно и то же. Кладовщику не нужно проектировать тесты качества, а Лене не нужно объяснять, что такое ИИ. Грамотность — это четыре уровня, и каждый человек в компании должен быть на своём.

Вот четыре уровня ИИ грамотности:

1. **Понимание.** Через него проходят все — от водителя до директора. На выходе человек понимает, что ИИ может и чего не может, видит возможности в своей работе, не боится. В «ФудПро» это сделали за неделю: четыре воркшопа по два часа, каждый сотрудник попробовал ИИ на десятке реальных задачах из своей работы. 20% теории и 80% практики. После воркшопа сорок процентов сказали «хочу узнать больше», десять процентов остались скептиками. Скептики — это нормально, давление ни к чему.
2. **Применение.** Менеджеры по продажам, кладовщики, все кто использует агента каждый день. На выходе человек знает, когда доверять агенту, когда проверять, умеет

давать конкретную обратную связь. Не «ваш ИИ врет», а «агент ошибся, потому что не учёл, что этот клиент не заказывает по пятницам». В «ФудПро» — тренинг на реальных кейсах: вот клиент, вот что предложил агент, вот где он ошибся, как бы вы дали обратную связь? Четыре часа в неделю, четыре недели. Время обучения выделенное. Оно снимается с других задач.

3. **Проектирование.** Лена, Игорь — те, кто строит правила, пишет тесты, читает метрики, решает, когда повышать автономию. На выходе человек проектирует свод правил, создаёт тесты качества, принимает решения на основе сводки. В «ФудПро» Лена дошла до этого уровня за три месяца практики с реальным агентом. Игорь — за два, потому что к тому времени уже видел результаты Лены и учился быстрее.
4. **Системное мышление.** Тот, кто видит всю картину: выбирает процессы для перестройки, оценивает барьеры, решает, куда направить следующий маховик. В «ФудПро» на этот уровень первой вышла Лена. Не потому что ей дали должность, а потому что через год практики она стала видеть следующие процессы сама: помогла Игорю запустить склад, предложила директору третий процесс — маршрутизацию доставки. Директор принимает решения о бюджете и приоритетах, но архитектуру проектирует Лена. Уровень определяется практикой, а не иерархией.

Уровень	Кто	Что умеет на выходе	Формат
Понимаю	Все (100%)	Видит возможности, не боится	Воркшоп, 4+ часа
Работаю	Пользователи агентов (60–80%)	Работает с агентом, даёт обратную связь	Тренинг, 4 ч/нед × 4 нед
Проектирую	Владельцы процессов (10–20%)	Пишет правила, тесты, читает метрики	Проект на реальном процессе, 2–3 мес
Вижу систему	Те, кто дорос практикой (3–5%)	Выбирает процессы, проектирует маховик	Практика на реальных проектах, 6–12 мес

Что не работает

Однодневный тренинг даёт иллюзию прогресса. Навык, который не закреплён практикой за четыре недели, исчезает. Через полгода даже после хорошего обучения навыки устаревают: новые модели, новые инструменты, новые паттерны.

Ссылка на онлайн-курс и «учись в свободное время» работает ещё хуже. Свободного времени у операционных сотрудников нет. Через три месяца курс прошли три процента.

Обучение, оторванное от процесса, бесполезно. «Напиши промпт для стихотворения» — сотрудник не видит связи со своей работой. Обучение работает, когда привязано к реальному процессу: вот ваши реальные клиенты, вот что агент предлагает, вот как это влияет на ваши показатели.

Главное правило: выделенное время. Четыре часа в неделю, снятые с задач, с конкретной программой. Без выделенного времени обучение падает до нуля к третьей неделе — рабочие задачи всегда побеждают. Компании, которые выделяют время на обучение, в четыре раза чаще добиваются результатов от ИИ-трансформации.

Обучить и забыть

Программа проведена, бюджет освоен, HR отчитался. Через полгода навыки устарели. Сотрудники работают по привычкам полугодовой давности.

В «ФудПро» обучение стало ритмом, а не событием. Раз в месяц — двухчасовая сессия обновлений для всех уровней: новые инструменты, новые кейсы, обмен опытом между процессами. Лена рассказывает Игорю, что сработало с клиентами. Игорь рассказывает Лене, что сработало на складе. Перекрёстное опыление, которое невозможно получить из онлайн-курса.

Люди обучены, процессы запущены, маховик крутится. Но кто за что отвечает? Кто владеет базой знаний? Кто пишет тесты? Кто решает, когда повысить автономию? Без ясных ролей ответственность размывается.

Следующая глава — роли.

Роли: кто за что отвечает

Маховик крутится, Лена довольна. Но через три месяца она замечает: всё замыкается на ней. Базу знаний обновляет она. Тесты пишет она. Метрики смотрит она. Решения об автономии принимает она. Когда Лена уехала в отпуск на неделю, никто не обновил базу знаний, никто не посмотрел сводку, два теста сломались — и никто не заметил.

Маховик без ясных ролей — это маховик, который держится на одном человеке. А один человек болеет, уходит в отпуск и выгорает.

Три роли

Новые задачи породили новые роли. Они правда нужны — появилась работа, которую раньше никто не делал.

Владелец процесса

Человек, который смотрит на маховик целиком. Читает метрики, решает, что улучшать, когда повышать автономию. Владеет базой знаний, сводом правил, тестами. В «ФудПро» это Лена для клиентского процесса и Игорь для склада. Владелец процесса не пишет код и не настраивает модели. Он смотрит на сводку и решает: «Частота заказов у нерегулярных клиентов падает — надо обновить базу знаний» или «Доля вмешательств стабильна два месяца — пора обсуждать следующий уровень автономии».

Откуда берётся: из руководителей, которые знают процесс изнутри. Лена была лучшим менеджером по продажам. Игорь двенадцать лет управлял складом. Таких людей не найти на рынке — они вырастают внутри. Путь от руководителя до владельца процесса — три-шесть месяцев практики с реальным агентом.

На старте пилота владелец процесса часто совмещает роль эксперта: Лена сама записывала правила, сама разбирала кейсы, сама обновляла базу знаний. Это нормально для маленькой команды. По мере роста процесса роли расходятся: Лена фокусируется на метриках и автономии, а эксперты берут на себя ежедневную работу с базой знаний.

Эксперт

Тот, кто знает предметную область до деталей и учит этому систему. Раньше он «знал сам», теперь «учит агента». В «ФудПро» это опытные менеджеры, которые знают, что шеф «Тополя» не берёт свинину, а «Берёзка» не пробует новинки. И кладовщики, которые знают, что в пятницу заказывать вдвое больше молочки. Четыре-шесть часов в неделю: разбор расхождений, обновление правил, проверка решений агента в сложных случаях.

Главный переход для эксперта — от информационной монополии к роли учителя системы. Петрович двадцать лет развозил продукты и знал каждого клиента в лицо. Раньше это знание жило только в его голове. Теперь оно в базе знаний — и работает, даже когда Петрович на маршруте.

Разработчик

Не только «пишет промпты», а скорее создает систему: как агент получает данные, как

принимает решения, как откатывается при ошибке. В компании на пятьдесят человек, как «ФудПро», это внешний подрядчик или один технический специалист. В компании на пятьсот — отдельная команда. Важно то, что эта роль обслуживает процесс, а не владеет им. Решения принимает владелец процесса, а не инженер.

Старые роли не исчезают

Менеджер по продажам в «ФудПро» раньше сам звонил клиентам, собирал заказы, помнил предпочтения. Теперь агент готовит предложения, а менеджер работает со сложными случаями: новые клиенты, конфликты, нестандартные запросы. Зона ответственности сместилась от рутины к суждению.

Кладовщик раньше прикидывал заказы на глаз. Теперь проверяет прогноз агента и корректирует. Работа стала точнее и интереснее — ушла самая нудная часть.

Здесь важно быть честным. Не каждый из двадцати менеджеров станет владельцем процесса. Реалистичная картина через год-два: трое-четверо перешли в новые роли, пятеро-шестеро работают с агентами в своих задачах, двое-трое ушли на другие направления, которые раньше не могли запустить. Замалчивать это — верный способ потерять доверие.

Карта перехода

Самый тревожный вопрос трансформации: «А что будет со мной?» Пока у человека нет ответа, он сопротивляется — и правильно делает.

Карта перехода — одностраничный документ для каждой роли, которая меняется. Пять вопросов.

Вопрос	Пример: менеджер по продажам «ФудПро»
Что я делаю сегодня?	Звоню клиентам, собираю заказы, веду CRM
Что заберёт агент?	Подготовка предложений, напоминания, стандартные заказы
Что останется за мной?	Новые клиенты, сложные случаи, обратная связь агенту
Чему мне научиться?	Работа с агентом, оценка его предложений, обратная связь
Сколько времени на переход?	Месяц тренинга (4 ч/нед) + месяц практики с наставником

Карта перехода не обещает, что всё будет как раньше. Она показывает конкретный путь: вот что изменится, вот что останется, вот что нужно освоить, вот сколько это займёт. Человек видит маршрут, а не туман.

Как понять, что переход состоялся? Только практикой. Человек месяц работает в новой роли и справляется: метрики его участка не упали, коллеги подтверждают, что он принимает решения сам. Если не справляется — ещё два месяца с наставником и пересмотр карты.

Главная ошибка: написать красивые карты перехода и положить в ящик. Карта без бюджета на обучение, без выделенного времени и без наставника — бумажка для отчёта, не план.

Роли распределены, люди обучены, маховик крутится. Осталось одно — начать. Но не через полгода, когда «будут готовы данные». В понедельник.

Часть 4. Старт

Что делать в понедельник

Что делать в понедельник

Книга прочитана. Маховик, пять шестерёнок, уровни автономии, тесты, метрики, роли — всё на месте. Осталось самое сложное: начать. Давайте поговорим о том, как сделать четыре конкретных шага на этой неделе.

Четыре шага

Первый шаг: выбрать один процесс.

Не берите самый простой. Возьмите самый болезненный, где высокий объём операций, понятный результат и конкретная боль. В «ФудПро» это были клиентские предложения: двадцать менеджеров, четыреста клиентов, текучка, знания уходят с людьми. Пройдитесь по трём-пяти кандидатам и для каждого ответьте на три вопроса:

- Есть ли данные, на которых агент может учиться?
- Можно ли измерить результат?
- Что будет, если агент ошибётся — поправимо или катастрофа? Процесс, где данные есть, результат измерим, а ошибка поправима — ваш первый кандидат.

Результат к пятнице: один процесс, выбранный по данным, а не по политическим соображениям.

Второй шаг: записать двадцать решений

Не ждите идеальной системы. Откройте таблицу — хоть в экселе — и начните записывать решения, которые принимаются в этом процессе прямо сейчас.

Шесть полей:

1. что на входе,
2. какие варианты,
3. что решили,
4. почему,
5. насколько уверены, что получилось.

Последнее шестое поле заполните через неделю. В «ФудПро» Лена за первую неделю записала двадцать пять решений по клиентам. Через месяц их было триста — и в них уже были видны паттерны, которые ни один менеджер не замечал по отдельности.

Результат к пятнице: таблица с двадцатью решениями. Это сырьё для базы знаний и первых тестов — две шестерёнки маховика запускаются одновременно.

Третий шаг: завести три самых базовых метрики.

Например, доля вмешательств (показывает зрелость процесса), одна метрика удержания (для «ФудПро» — частота повторных заказов) и одна финансовая (средний чек, стоимость операции — зависит от процесса). Для каждой запишите текущее значение, цель через три месяца и метрика-противовес. Это ваш контрольный дашборд.

Результат к пятнице: одна страница с тремя метриками. Базовый уровень, цель, противовес.

Когда что-то идёт не так

Агент ошибается слишком часто. Через месяц работы точность ниже порога, четверть решений мимо. Не повышайте автономию — это сигнал, что база знаний не готова. Возьмите двадцать самых частых ошибок, добавьте правила, прогоните тесты через две недели. На первом уровне автономии цена ошибки минимальна: решает человек.

Качество падает после повышения автономии. Агент отлично работал, когда человек проверял каждое решение. Дали больше свободы — и ошибки полезли. Откатитесь на уровень назад. Это не провал, это нормальный механизм маховика. Автономию можно понижать так же легко, как повышать. Найдите паттерн ошибок в логе решений, дополните свод правил, наберите данные заново.

Люди не участвуют. Никто не заполняет таблицу решений, эксперт «не находит времени» для базы знаний, процесс замирает. Вернитесь к пяти рычагам: новая роль, стимулы, ответственность, быстрая победа, выделенное время. Без конкретного владельца процесса маховик не стартует.

Если через три месяца нет сдвига ни в одной из трёх метрик — процесс выбран неправильно. Это нормально. Возьмите следующего кандидата. Выбор процесса важнее качества исполнения.

Что дальше

К концу первой недели у вас есть: процесс, двадцать записей, двадцать тестов и три метрики. Маховик ещё не крутится — но шестерёнки на месте.

Дальше — подключите агента в режиме тени: он обрабатывает задачи параллельно с людьми, но решения не применяются. Сравните результаты с вашими тестами. Если точность выше

порога — запускайте канарейку на десять процентов. Дальше — по таблице переходов из главы про автономию.

Через три месяца оглянитесь. Таблица решений выросла до тысячи записей. Тесты расширились. Метрики показывают динамику. Маховик начал крутиться. И главное — вы видите следующий процесс.

А если чувствуете, что для запуска не хватает конкретных навыков — следующая глава про это.

От теории к практике

Если вы дочитали до сюда — вы понимаете, что должно получиться. Но наверняка остались конкретные вопросы: «а как именно это сделать у нас?», «а что если...», «а мы пробовали, не взлетело». Если хотите обсудить — [пишите](#), всегда рады поговорить! Особенно ценим критику: если что-то не сработало или звучит наивно — скажите, это делает методику лучше.

Вот какие навыки вам пригодятся для старта:

Найти процесс	Собрать базу знаний	Спроектировать агента	Собрать прототип	Запустить пилот	Метрики и экономика
Выбор процесса-кандидата	RAG и контекст-инженерия	Архитектура и интеграции	Вайб-кодинг	Пилот и мониторинг	Доля вмешательств, контрметрика
Декомпозиция на цепочки решений	Чанкинг регламентов и данных	Оркестрация цепочек	Промпт-инженерия для кода	Тесты до запуска	Unit-экономика процесса
Скоринг	Эмбединги и векторный поиск	Подключение через MCP	Отладка диалогом	A/B в режиме тени	ROI-модель
Стейкхолдер-анализ	Контекстное окно	Обработка ошибок	Границы метода	Мониторинг	Пороги перехода автономии
Формулировка гипотезы	Гибридный поиск	Уровни автономии	Тестирование без тестировщика	Работа с сопротивлением	Защита бюджета

Каждый навык осваивается практикой на реальном процессе с обратной связью. Опыт программирования не нужен. Именно так устроен курс «ИИ-чемпионы» от авторов книги: двенадцать недель от поиска процесса до работающего прототипа.

Для кого: руководители направлений, операционные директора, продуктовые менеджеры — для всех будущих владельцев процесса.

[Подобности и запись тут.](#)

Подпишитесь на [наш канал](#), чтобы не пропустить новые версии книги и кейсы.

Удачного понедельника!