

Внедрение ИИ в бизнес: что это на самом деле и как не получить кладбище прототипов

ИИ у вас уже есть. Он просто не работает.

Девять из десяти российских проектов внедрения генеративного ИИ откладываются на неопределённый срок — не потому, что технология слабая, а потому, что внедряют не то, не так и не с теми. Этот гид — про то, как отличить внедрение, которое поедет, от внедрения, которое умрёт. Без кода. С тем, что нужно знать владельцу, чтобы ставить задачу, принимать результат и не платить дважды.

Содержание

1	Что такое внедрение ИИ на самом деле	3
1.1	Это два разных дела	
1.2	Шесть способов применить ИИ	
1.3	Как задачи перетекают между способами	
2	Нужно ли это вашему бизнесу	5
2.1	Три вопроса на входе	
2.2	Где физически возникают деньги	
2.3	Улучшить процесс или перестроить бизнес	
3	Кому отдать задачу: программе, агенту или человеку	7
3.2	Восемь вопросов к шагу	
3.3	Примеры	
3.5	Паспорт задачи — 15 минут владельца	
4	Какие процессы улучшать первыми	11
5	Как внедрять: шесть шагов	13
6	Чеклист приёмки: «внедрено» против «навайбкодили»	15
7	Программисты и «вайбкодеры»: скорость и надёжность	16
8	Двенадцать заблуждений	18
9	С чего начать: две недели владельца	19
—	Откуда цифры	20

Как читать. Разделы 1–2 — язык и тест «нужно ли». Раздел 3 — правило, которым вы будете пользоваться чаще всего. Разделы 4–6 — порядок внедрения и приёмка. Раздел 7 — как совместить скорость и надёжность в команде. Разделы 8–9 — заблуждения, план на две недели и вопросы подрядчику.

РАЗДЕЛ 1

Что такое внедрение ИИ на самом деле

1.1 Это два разных дела

Под словами «внедрить ИИ» скрываются два несвязанных вида работы. Их путают чаще всего — и именно из этой путаницы растут ложные ожидания.

	ИИ строит	ИИ работает
Что делает модель	пишет программу под вашу задачу — за дни, а не месяцы	выполняет шаг процесса каждый день, каждый раз заново
Что вы получаете	обычную программу: работает одинаково, без ИИ внутри	«сотрудника», который читает, пишет, решает
Сколько стоит	один раз, при создании	за каждый прогон, всегда
Насколько предсказуемо	полностью: сегодня и через год — одно и то же	вероятностно: может ошибиться, как человек
Пример	учёт денег, печать этикеток, сверка выплат маркетплейса	разбор входящей почты, черновик ответа клиенту, объяснение «почему упала маржа»

Следствие для владельца: «нам сделали финансовое приложение за месяц» и «у нас агент отвечает на отзывы» — разные продукты, с разной экономикой и разными рисками. Когда подрядчик говорит «внедрим ИИ», первый вопрос: **какой из двух?**

1.2 Шесть способов применить ИИ

Чтобы не спорить о словах — одна шкала. Сверху вниз: от дешёвого и предсказуемого к гибкому и дорогому.

№	Способ	Как это выглядит	Когда уместно
0	Не трогать / убрать шаг	процесс меняется так, что шаг больше не нужен	шаг редкий и дешёвый, или его результат никто не использует
1	Помощник	сотрудник работает с ИИ в диалоге: исследование, разовый анализ, черновик	задача разовая или каждый раз новая
2	Регламент	повторяемая процедура, упакованная в одну команду; запускает человек	повторяется, но не настолько часто, чтобы строить программу
3	Программа («станок»)	ИИ написал программу; она работает без ИИ — одинаково, быстро, почти бесплатно	часто, одинаково, ошибка дорогая; все «канонические цифры»
4	Конвейер с ИИ-шагом	программа ведёт процесс, а ИИ делает один шаг: прочитать документ, написать комментарий	вход или выход — свободный текст, а правило в середине чёткое

№	Способ	Как это выглядит	Когда уместно
5	Агент («сотрудник»)	ИИ сам решает, какие шаги сделать и какими инструментами	шаги заранее не известны, ошибка дешёвая или её утверждает человек

Метафора, которая помогает не ошибаться: **программа — станок, агент — сотрудник**. Станок долго проектируют под одну операцию, зато потом он делает её тысячи раз в день идеально и бесплатно. Сотрудник справится и с новой задачей, и с нестандартной, — но каждое его действие стоит денег, и он может ошибиться. Поручать станку новое — невозможно. Поручать сотруднику крутить одну гайку десять тысяч раз в день — расточительно и опасно.

«Перевести всё на агентов» — это уволить конвейер и посадить людей крутить гайки. Первые дни будет казаться, что работает. Потом придут счета за каждую операцию и первая дорогая ошибка в платёжке.

1.3 Как задачи перетекают между способами

В здоровой системе задачи движутся снизу вверх по шкале — от гибкого к закреплённому:

новое	помощник: сделали руками с ИИ
повторилось	регламент: упаковали в команду
стало частым	программа: закрепили в код
агент	больше не делает это сам — пользуется программой как инструментом

Агенты — исследовательский отдел: обрабатывают новое и нестандартное. Программы — производство: эксплуатируют найденное и проверенное. Бонус момента: ИИ сам умеет писать программы, поэтому «закрепить» теперь стоит дни, а не месяцы. Но решение, что закреплять, — управленческое. Его принимаете вы.

РАЗДЕЛ 2

Нужно ли это вашему бизнесу

2.1 Три вопроса на входе

Ответьте честно, прежде чем звать кого-либо.

- 1. **Что в компании делается руками, и сколько это стоит в месяц?** Не «много», а в часах и рублях: «менеджер каждый понедельник два часа сводит выкупы по кабинетам». Если посчитать не получается — с этого и начинается работа.
- 2. **Кого вы собираетесь нанимать в ближайшие полгода и почему ещё не наняли?** Это главный вопрос. Внедрение даёт деньги только там, где рост упёрся в людей.
- 3. **Кто в компании глубже всех понимает процесс, который болит?** Без такого человека схема не собирается: некому быть автором изменений.

2.2 Где физически возникают деньги

Освобождённый ресурс превращается в прибыль ровно двумя путями.

Путь первый — сократить людей. Работает арифметически, не работает практически. В малом и среднем бизнесе сотрудники — свои люди; владелец не увольняет, а перераспределяет. Вдобавок именно эти сотрудники должны внедрять систему, которая их сокращает. Сопротивление здесь не иррационально — оно рационально. Это и есть причина, по которой девять из десяти проектов не доезжают: ИИ навязывается сверху, без вовлечения тех, кто им пользуется.

Путь второй — расти без найма. Тот же штат обслуживает вдвое больший объём. Экономия возникает в несостоявшихся расходах: пять человек, которых не пришлось нанять. Никого не увольняют, сопротивления нет, эффект считается однозначно.

Бизнес без роста автоматизацию не оплатит — он получит комфорт, а не деньги, и через полгода откажется. Бизнес, который прямо сейчас упёрся в наём, оплатит охотно: предложение звучит как «не нанимай пятерых», а не «уволь пятерых».

Цифра для трезвости: в России закрывается не более 55–60 % вакансий. Почти половина не закрывается вообще. «Рост упёрся в наём» — не предположение, а статистика.

2.3 Улучшить процесс или перестроить бизнес

Это тоже два разных продукта, и путать их дорого.

	Улучшить существующий процесс	Перестроить «как с нуля» (AI-native)
Что делаем	ускоряем то, что уже есть	проектируем процесс так, как построили бы сегодня
Роль человека	в середине, ему помогают	на краю: решения и исключения
Типичный эффект	10–25 %	кратный

	Улучшить существующий процесс	Перестроить «как с нуля» (AI-native)
Что ломается	ничего	роли, иерархия, привычный порядок работы
Кто соглашается	почти все	почти никто с первого раза

Попытка начать с перестройки — главный способ попасть в те самые 90 % отложенных: она требует ломать сложившееся до того, как появилось доверие. Рабочая последовательность: быстрый измеримый результат на существующем процессе → доверие и доступ к данным → предложение перепроектировать. **Перестройка — второй контракт, не первый.**

РАЗДЕЛ 3

Кому отдать задачу: программе, агенту или человеку

3.1 Правило разбирает не процесс, а шаг

Самая частая ошибка масштаба — «автоматизировать процесс агентом». Процесс почти всегда смешанный: часть шагов — чистая механика, часть — суждение, часть — ответственность человека. Поэтому сначала процесс раскладывают на шаги (это делается за 20 минут на доске), а потом каждому шагу задают вопросы ниже.

3.2 Восемь вопросов к шагу

Вопросы 3 и 8 — ворота: «нет» на любой из них останавливает разбор, и задача остаётся у человека. Остальные — взвешивание: они выбирают между программой, конвейером, агентом и помощником. Вопросы 6 и 7 — последствия: что пойдёт не так у выбранного варианта и кто это утверждает.

1. Нужен ли этот шаг вообще?

Если результат никто не использует — убрать. Если шаг можно не делать, изменив соседний (не сверять, а получать данные так, чтобы сверять было нечего), — убрать. Лучшая автоматизация — та, которой нет.

2. Как часто и что стоит?

Реже раза в неделю и дешевле получаса — не трогать: разработка и поддержка не окупятся никогда. Это самая частая и самая дорогая ошибка любителей: автоматизируют то, что делается раз в месяц за двадцать минут, и обнаруживают это через полгода.

3. Есть ли пример правильного результата?

Нет эталона — нечего проверять и нечего делегировать. Пока нет примера «вот так правильно», задача остаётся у человека с помощником.

4. Можно ли записать правило полностью?

Проверка простая: два независимых человека с этим правилом дают одинаковый результат. Если да — **программа**, и никакого ИИ внутри. Если правило чёткое, но вход — письмо, документ, фото — **конвейер с ИИ-шагом**: ИИ читает, программа проверяет и исполняет. Если правило не записывается и нужно суждение — дальше.

5. Насколько дёшево проверить результат?

Автоматически — агент может работать без человека. Человек за десять секунд (черновик перед отправкой) — агент с утверждением. Только эксперт и долго — ИИ помогает эксперту, а не заменяет его.

6. Какова цена ошибки и можно ли её отменить?

Деньги, документы, отчётность, публикация, всё необратимое — **исполняет программа, утверждает человек, агент только готовит**. Черновик, который всё равно перечитают, — можно агенту.

7. Что система будет менять снаружи?

Только читает — агенту можно сразу. Только уведомляет — агенту с лимитом расходов и журналом. Пишет в кабинет маркетплейса, платит, отправляет клиенту — только через подтверждение человеком. Снимать подтверждение — по счёту, а не по ощущению: если допустима одна ошибка на десять — после 30 прогонов подряд без правок; одна на сто — после 300. Любая правка обнуляет счёт. Промежуточный режим — право вето: система делает через сутки, если вы не остановили.

8. Данные есть, и они доступны?

Если данные в головах, чатах и разрозненных таблицах — сначала порядок в данных, потом ИИ. Агент поверх хаоса — генератор уверенных ошибок. Если есть персональные данные — российский контур модели или обезличивание до отправки. Всё, что подписывается или лицензируется, — никогда полностью автоматически.

3.3 Примеры

Задача	Кто делает	Почему
Провести платёж, списать остаток со склада	программа	часто, одинаково, ошибка дорогая
Выставить счёт по шаблону	программа	реквизиты не терпят «творчества»
Сверить выплаты маркетплейса с учётом до единицы	программа; расхождения — человеку	цифры каноничны
Ежедневный отчёт по продажам	программа считает, ИИ пишет «почему изменилось»	цифры — канон, интерпретация — суждение
Извлечь реквизиты из счёта в письме, создать заявку на оплату	ИИ читает → программа проверяет по справочнику → человек согласует	вход свободный, правило чёткое, платёж необратим
Ответить на отзыв или нестандартное письмо	агент готовит → человек утверждает → программа отправляет	нужен контекст; отправка необратима
Разобрать 200 писем с утра	агент	рутина с суждением, ошибка дешёвая: письмо можно перечитать
Напомнить должникам об оплате	программа находит просроченных → ИИ пишет текст → человек утверждает → программа отправляет	комбинация: суждение + необратимое действие
Изменить ставки в рекламе	программа считает ставку от вашей экономики; меняет — только после «да»; позже — сама в заданных границах	деньги; доверие накапливается прогонами
Разовый анализ «почему упали продажи в марте»	помощник	вопрос задан один раз — разработка не окупится никогда

Задача	Кто делает	Почему
Ночное исследование «изучи конкурентов»	агент	шаги заранее не известны, только читает

ИИ думает — программа исполняет — человек утверждает необратимое.

Это самый частый узор. Агент никогда не держит «кнопку»: письмо отправляет программа по утверждённому черновику.

3.4 Три правила, которые не обсуждаются

- **Канонические цифры считает только программа.** Выручку, остатки, долги, маржу ИИ читает из системы и объясняет — но не вычисляет.
- **Всё, что делает агент, — черновик,** пока вы явно не решили иначе. Действие — отдельный шаг с отдельным решением.
- **Сначала агент, потом программа.** Новую задачу агент делает «руками» под присмотром; когда шаги стабилизировались (три одинаковых прогона без правок) — закрепляем в программу.

3.5 Паспорт задачи — 15 минут владельца

Ни одна задача не должна уходить в работу без этих ответов. Заполняет владелец процесса на своём языке; техническая часть — дело исполнителя.

ЧТО ВХОДИТ

Задача: что должно происходить — глагол и объект
 Откуда данные: система / файл / письмо / «у кого-то в голове»
 Как часто: N раз в день / неделю / месяц

ПО КАКИМ ПРАВИЛАМ

Правило: можно ли записать полностью — так, чтобы двое дали один ответ
 Что будет при ошибке: потеряем деньги / документ / репутацию / ничего страшного
 Ограничения: сроки, лимиты, персональные данные, что подписывается

ЧТО ВЫХОДИТ

Пример правильного результата: ссылка или файл — обязательно
 Что система меняет снаружи: только читает / уведомляет / пишет, платит, отправляет

КТО ИСПОЛНЯЕТ

Владелец: кто в компании отвечает за результат (не подрядчик)
 Известные исключения: те 20 % случаев, которые «не как обычно»
 Кто разбирает исключения: имя и сколько минут в день у него на это есть

Сколько стоит сейчас: часы в месяц × ставка = ₽/мес

Четыре группы — четыре стороны любой функции: что входит, по каким правилам, что выходит, кто исполняет. Ответ на вопрос «кому отдать» — это выбор исполнителя при записанных правилах; если правила записать нельзя, исполнителя выбирать рано.

Если паспорт не заполняется за 15 минут — это не проблема формы, а сигнал: процесс не понятен даже его владельцу. Автоматизировать непонятое нельзя.

РАЗДЕЛ 4

Какие процессы улучшать первыми

4.1 Карта по двум осям

	ОДИНАКОВО КАЖДЫЙ РАЗ	КАЖДЫЙ РАЗ ПО-РАЗНОМУ
ЧАСТО	программа станок: одна операция тысячи раз, бесплатно и без ошибок	агент или конвейер нужно суждение на каждом прогоне; необратимое — через человека
РЕДКО	не трогать стоимость создания и поддержки выше выигрыша — здесь любители теряют больше всего	человек с помощником разовое, экспертное, каждый раз новое

Левый нижний угол — главное открытие для большинства компаний. Редкое и одинаковое автоматизировать не надо: стоимость создания и поддержки выше выигрыша. Именно здесь любители теряют больше всего.

4.2 Критерии первого процесса

Первый процесс — не самый впечатляющий, а самый выгодный для доверия:

- измеряется в деньгах: часы × ставка, а не «станет удобнее»;
- есть владелец внутри компании — человек, которому это нужно;
- данные существуют и доступны;
- цена ошибки терпима: черновики, отчёты, напоминания — не платежи;
- результат виден за 2–4 недели.

Автоматизируйте узкое место, а не громкое.

Два быстрых счёта отсекают «громкие» кандидаты. Первый — узкое место: автоматизируйте шаг, где работа ждёт людей; ускорение любого другого шага не даёт ничего — очередь просто переезжает. Второй — доля времени: если шаг занимает пятую часть времени всего процесса, автоматизация даже до нуля ускорит процесс максимум на четверть.

Правильные первые кандидаты — ежедневная рутина с дешёвой ценой ошибки: сводки, черновики ответов, регулярные отчёты, напоминания, извлечение данных из документов. Быстрый видимый результат, безопасные ошибки, команда привыкает.

4.3 Пример карты для торговой компании на маркетплейсах

Направление	Сразу (дни)	Потом (недели)
Отчётность	сбор отчётов площадок по API в одну таблицу, ежедневная сводка	витрины данных, история, сверка с площадкой до единицы
Реклама	утренняя сводка по кампаниям, расчёт целевой ставки от экономики	агенты, которые меняют ставки и чистят минус-фразы по правилам
Финансы	выгрузка выплат, юнит-экономика и P&L по SKU	внутреннее приложение: заявки на оплату, бюджет, сверка с банком и 1С
Склад и закупки	что заканчивается, что зависло, оборачиваемость	прогноз поставок, автозаявки
Контент	карточки, описания, ответы на отзывы — с подтверждением	контент-конвейер по расписанию
Люди	один сотрудник-«чемпион» осваивает инструмент за сессию	свои команды и агенты под каждую роль

РАЗДЕЛ 5

Как внедрять: шесть шагов

Шаг 0. Аудит кладбища прототипов

Если в компании уже что-то пробовали — это первый шаг. По каждому прототипу фиксируется: работает или нет, кто владелец, на каких данных, что именно ломается. Часть прототипов чинится за часы — это быстрые победы и основание для доверия.

Четыре вопроса к каждому мёртвому прототипу — и к каждому парный «а что при этом работает»: что именно не работает (и что работает), где (и где нет), когда началось (и когда было нормально), в каком объёме (и в каком нет). Разница между «есть» и «нет» указывает на причину из таблицы точнее, чем спор.

Причина «навайбкодили, а не работает»	Как проявляется
Данных нет или они грязные	работает на демо, падает на реальных
Нет владельца	сделали и бросили, никто не пользуется
Не обработаны исключения	80 % случаев проходит, 20 % ломают процесс
Нет интеграции	данные надо вводить руками, и никто не вводит
Нет мониторинга	сломалось — никто не заметил
Доступы и безопасность	30–40 % ИИ-кода содержит уязвимости
Эксплуатация дороже пользы	никто не посчитал
Непередаваемость	поддерживать может только автор

Шаг 1. Карта процессов

Все повторяющиеся процессы — на карту из раздела 4: что в программу, что агенту, что человеку с помощником, что не трогать.

Затем — выжать, а не строить: на узком месте сначала убрать лишние шаги и дать людям простой регламент. Половина автоматизаций после этого оказывается не нужна, а оставшиеся строятся на чистом процессе, а не на хаосе.

Шаг 2. Первый процесс и базовая линия

Выбор по критериям 4.2. До начала работ — замер: сколько часов и денег процесс стоит сейчас. Без базовой линии потом нечего будет предъявить.

Шаг 3. Контур эксплуатации

Всё, что отличает прототип от работающей системы, — чеклист в разделе 6. Это то, за что платят, и то, чего нет ни у одного «сделали на выходных».

Шаг 4. Замер

Базовая линия до, замер после. Два счётчика, не требующие спора «а чья это заслуга»: **сколько человек не наняли и сколько стоила бы такая разработка на рынке.**

Эффект — одно из трёх: компания делает больше тем же штатом; не понесла расходы, которые собиралась; не заказала разработку, которую пришлось бы купить. Если не произошло ничего из трёх — это комфорт, а не автоматизация, и через полгода от него откажутся.

Шаг 5. Повтор

Следующий процесс по той же схеме. Методология переиспользуется, предметная часть — нет.

Ритм, который работает: 30 / 60 / 90.

30 дней — замеренная базовая линия; 60 дней — процесс с реальным владельцем внутри компании; 90 дней — документированная экономия или прирост выручки, отнесённые к проекту.

РАЗДЕЛ 6

Чем «внедрено» отличается от «навайбкодили»: чеклист приёмки

Задача считается внедрённой, когда закрыт каждый пункт. Проверить их можно, не понимая ни строчки кода, — достаточно задать вопрос и потребовать показать.

для ЛЮБОЙ АВТОМАТИЗАЦИИ

- ☐ источник данных назван, качество известно; грязные данные обрабатываются, а не роняют процесс;
- ☐ исключения (те самые 20 %) описаны и уходят человеку с контекстом, а не теряются;
- ☐ исключения к человеку — тоже очередь: посчитано, сколько случаев в день и сколько минут на каждый, и это не больше 80 % времени того, кто разбирает; при большей загрузке очередь растёт быстрее, чем кажется — при 90 % ждать в девять раз дольше, чем при 50 %;
- ☐ агент не выпускает больше, чем вы успеваете разбирать: у него есть дневной лимит, привязанный к мощности того, кто утверждает;
- ☐ понятно, что происходит при отказе: что видит пользователь, кто узнаёт, как повторить;
- ☐ есть мониторинг и оповещение: сломалось — заметили в тот же день;
- ☐ доступы минимальные, пароли и ключи не в переписке и не в коде; персональные данные — в российском контуре или обезличены;
- ☐ владелец внутри компании умеет запустить, проверить и остановить;
- ☐ стоимость эксплуатации посчитана против пользы: серверы, оплата моделей, часы контроля;
- ☐ передаваемость: поддерживать может не автор — есть описание, паспорт, инструкция.

дополнительно для АГЕНТОВ и ИИ-ШАГОВ

- ☐ лимит расходов на каждый ключ: агент физически не потратит больше N;
- ☐ инструменты только из утверждённого списка, с уровнем доступа «читать / уведомлять / писать»; «писать» — через очередь подтверждений;
- ☐ журнал внешних действий: кто, что, по чьему утверждению;
- ☐ эталонный набор проверок, который перепрогоняется при смене промпта или модели — чтобы улучшения не ломали то, что работало;
- ☐ выключатель: остановить одним действием, без программиста.

К каждому пункту — три вопроса: как предупреждаем, что делаем, если всё-таки случилось, и как узнаем, что случилось. Пункт без ответа на третий вопрос не закрыт: о поломке вы узнаете от клиента.

Если хотите «всегда включённого» ИИ-сотрудника на своём сервере (такие системы уже есть — например, Hermes Agent, работающий круглосуточно, с памятью и подключением к мессенджерам), до установки должно быть верно всё то же самое — плюс одно: **навыки, которые агент «научился» сам, — это код без проверки.** Они попадают в постоянный набор только после просмотра человеком, иначе вы получите кладбище прототипов внутри агента.

РАЗДЕЛ 7

Программисты и «вайбкодеры»: как получить и скорость, и надёжность

7.1 Спор, который не решается на каждой задаче

Бизнес требует «вчера», и ему всё равно, как сделано. Программист требует «правильно», потому что видел, как рассыпается. Оба правы. Сегодня к этому добавился третий участник — сотрудник без программистского образования, который делает работающие прототипы с помощью ИИ. В России уже сформировалась целая категория таких «вайбкодеров», и больше половины из них пишут код без навыков программирования.

Чем это оборачивается без правил — измерено на сотнях тысяч коммитов: более 15 % ИИ-кода вносит хотя бы одну проблему, и каждая пятая из них так и остаётся неисправленной; ИИ-код даёт в 1,7 раза больше проблем на изменение; технический долг после внедрения ИИ-инструментов растёт на 30–41 % в год. Код выглядит правильным и создаёт ложную уверенность.

Развернём спор. Цель у обоих одна — работающая система. Бизнесу для неё нужна скорость, инженеру — надёжность. Отсюда два требования: «пусть специалист делает сам» и «только через инженера». Конфликт живёт не в цели и не в потребностях — только в требованиях, и снимается одним ходом, который даёт обе потребности сразу.

Вывод не в том, чтобы запретить. Вывод в том, что **спор «быстро или правильно» решается не на каждой задаче, а один раз — рельсами.**

7.2 Что такое рельсы

Рельсы — правила, зашитые в инструменты так, что их нельзя нарушить случайно. Три уровня:

Уровень	Что это	Кто может обойти
Мягкий	правила проекта, которые ИИ-помощник читает в начале каждой сессии: какой стек, как устроены папки, что запрещено	любой — потому и мягкий
Средний	обязательный порядок работы: замысел → короткая спецификация → план → код → проверки → доставка в «песочницу»; нельзя писать код без утверждённого описания	тот, кто не пользуется инструментами
Жёсткий	автоматические проверки при каждом изменении; в «боевую» систему — только через проверку инженером; запреты на уровне базы данных; ключи и пароли только в защищённом хранилище	никто без инженера

Правило: **всё обязательное — на жёстком уровне**. Мягкое и среднее — для удобства и скорости, не для безопасности. Уговоры не работают; работает то, что исполняется само.

7.3 Кто что делает

	Ваш специалист-автор	Инженер — владелец рельсов	ИИ-помощник
Делает	описывает и собирает функции своей области; проверяет на своих данных; выкладывает в песочницу	строит и держит рельсы; проверяет всё, что идёт в боевую систему; отвечает за деньги, доступы, интеграции, базу	пишет код по правилам и описанию; на границах останавливается и спрашивает
Не делает	не трогает боевую систему напрямую, не добавляет чужие библиотеки, не пишет в чужие системы	не делает предметную работу за специалиста — иначе всё превращается в заказную разработку с её сроками и ценой	не обходит правила

Что специалист может делать сам: экраны, отчёты, справочники, расчёты, фильтры, выгрузки. Что только через инженера: движение денег, доступы и права, необратимые изменения базы, отправка наружу (кабинеты, платежи, рассылки), ключи и пароли.

7.4 Пример из практики: четыре захода, три мёртвых, один живой

Торговая компания на маркетплейсах. Четыре попытки построить одно и то же внутреннее финансовое приложение. Первые три остановились — включая ту, где предметные специалисты уже делали большую часть изменений сами. Четвёртая живёт и развивается: за месяц — 784 изменения, 48 моделей данных, 40 экранов, автоматические проверки и доставка в два контура, 43 % изменений внёс финансовый аналитик, а не программист. Промышленный контур доставки был готов в первый день. Рыночная альтернатива такой системы — заказная разработка от 3–8 млн ₽ и 7–13 недель только на базовый модуль.

Люди в мёртвом и живом заходах — те же. Технология — та же. Разница — рельсы: зафиксированный стек, запрет на самовольные библиотеки, обязательное описание до кода, единственный путь в боевую систему через проверку инженера, контур доставки с первого дня, терминология, понятная неразработчикам.

Разница между заходом, который умрёт, и заходом, который поедет, — не технология и не люди, а методология.

7.5 Что требовать от своей команды или подрядчика

- Правила проекта, спецификации и планы лежат в самом репозитории, а не в чате: это память проекта, которую читает и человек, и ИИ в каждой сессии. Именно поэтому быстрый результат получается production-ready: каждый раз старт с правильных рельсов, а не с чистого листа.
- Две среды: «песочница», где нельзя навредить, и боевая, куда — только через проверку.
- Автоматические проверки при каждом изменении и автоматический откат при неудаче.
- Ваши специалисты — авторы, инженер — рельсы. Если инженер начал делать предметную работу за специалиста — модель вырождается, и цена должна стать ценой разработки.

РАЗДЕЛ 8

Двенадцать заблуждений

Заблуждение	На самом деле
ИИ — это чат	чат отвечает текстом; инструмент работает с вашими файлами, кабинетами и базой: берёт данные, считает, делает отчёт, пишет и запускает программу
Внедрить ИИ — раздать сотрудникам доступ к нейросети	это помощник (способ 1), не внедрение; эффект есть, но он не измеряется и не масштабируется
Всё на агентов	счёт за каждую операцию и первая дорогая ошибка в платёжке; чёткое правило — в программу
Агенты — игрушка	штат людей на задачах, которые агент делает за копейки, и месяц разработки на каждый новый вопрос
Ограничений нет	ограничения есть, но не в модели: в данных, владельцах, исключениях и стоимости эксплуатации; кто верит в «нет ограничений», получает кладбище
Сначала платформа, потом процессы	месяцы инфраструктуры и ноль процессов в работе; один процесс до конца, платформа вырастает из повторов
Пусть ИИ считает деньги	«уверенно неправ»; цифры считает программа, ИИ объясняет
Автоматизировать всё, что делается руками	редкое и одинаковое не окупается никогда; а ускорение не узкого места не даёт ничего
Прототип работает — значит, внедрено	прототип и система отличаются пунктами чеклиста из раздела 6
Внедрим сверху, сотрудники привыкнут	причина девяти из десяти провалов; работает обратное — сотрудник становится автором
Экономия — это сокращение людей	в малом бизнесе не происходит; деньги — в росте без найма
Начнём с перестройки бизнеса под ИИ	это второй контракт; первый — измеримый результат на существующем процессе

РАЗДЕЛ 9

С чего начать: две недели владельца

1. **Список ручной работы с часами.** Каждый руководитель направления — три задачи, которые делаются руками и раздражают больше всего; по каждой — часы в месяц.
2. **Один эксперт-автор.** Человек, который глубже всех знает процесс и готов учиться. Не вся компания сразу — один «чемпион».
3. **Аудит кладбища.** Всё, что уже пробовали: работает или нет, кто владелец, что ломается.
4. **Первый процесс** — по критериям раздела 4.2, с заполненным паспортом из 3.5 и замеренной базовой линией.
5. **Чеклист приёмки** из раздела 6 — в договор или в задачу, до начала работ.

Вопросы любому подрядчику — и ответы, которые должны насторожить

Вопрос	Плохой ответ
Что из этого будет программой, а что — агентом, и почему?	«всё на ИИ», «всё на агентах»
Где пример правильного результата, по которому будем проверять?	«увидите — поймёте»
Кто в моей компании будет владельцем после сдачи?	«мы будем поддерживать» — и ничего про вашего сотрудника
Что происходит с теми 20 % случаев, которые не как обычно, — и кто успевает их разбирать?	пауза
Сколько это стоит в эксплуатации в месяц: серверы, модели, контроль?	«копейки» без цифр
Как я это остановлю без вас?	«зачем останавливать»
Что агент может менять снаружи, кто это утверждает и после скольких прогонов вы предложите утверждение снять?	«он сам разберётся»
Покажите не только успехи, но и заход, который умер, — и почему	только успехи

Последний вопрос — самый полезный. Тот, кто может объяснить, почему умерли его прошлые проекты, знает, чем отличается заход, который поедет.

ИСТОЧНИКИ

Откуда цифры

- Девять из десяти российских проектов внедрения генеративного ИИ откладываются; причина — ИИ навязывается сверху без вовлечения сотрудников — Computerworld Россия / OSP, Коммерсантъ, 2026.
- 90 % российских компаний застревают на пилотах ИИ-агентов — отраслевые обзоры, 2026.
- 71 % организаций используют ИИ-агентов, до production за год дошли 11 % сценариев — Camunda, State of Agentic Orchestration and Automation, 2026.
- Более 15 % ИИ-коммитов вносят проблему, 22,7 % остаются неисправленными; в 1,7 раза больше проблем на изменение — эмпирическое исследование 302 600 коммитов в 6 299 репозиториях (arXiv, 2026); техдолг +30–41 % в год — Sonar, Tembo, 2026.
- 52 % российских вайбкодеров пишут код без навыков программирования — RB.ru, 2026.
- Уровень закрытия вакансий 55–60 %, дефицит разработчиков 500–700 тыс. — данные рынка труда РФ, 2026.
- Стоимость заказной разработки корпоративной системы 3–8 млн ₽, базовый модуль 7–13 недель — публичные прайсы студий, 2026.
- Различие «workflow / agent» и принцип «сначала самое простое» — Anthropic, Building effective agents.
- «30 и 300 прогонов» — правило трёх из статистики (0 ошибок в n случаях → доля ошибок не выше 3/n с доверием 95 %); «узкое место» — теория ограничений; «пятая часть времени — не больше четверти ускорения» — закон Амдала; «при 90 % загрузки ждать в девять раз дольше» — теория очередей. Формулы приведены в упрощённом виде.
- «Выжать до того, как строить», лимит выпуска и эффект «одно из трёх» — теория ограничений (Голдратт); четыре группы паспорта — IDEF0; ворота и взвешивание, «есть / нет», «предупредить / отреагировать / узнать» — рациональные процессы Кепнера-Тrego. Все три — из практики автора.
- Пример из практики — обезличенные данные из репозитория; состав систем и сроки — по истории изменений, бизнес-эффект — со слов владельца.