

Экспертная мини-книга

Цифровая трансформация

Как увидеть технологический сдвиг раньше рынка и собрать дивиденды с ИИ

Константин Потапов

Senior Python Backend / R&D Engineer

От чатботов к управляемым агентным контурам, данным, проверкам и бизнес-результату

Карта книги

Об авторе	4
Часть I. Зачем ловить сдвиг	7
Глава 1. Как предугадывать будущее без магического шара	8
Часть II. От промпта к рабочему контуру	11
Глава 2. Почему “хороший промпт” больше не главное	12
Глава 3. Нулевая точка: просто модель	14
Глава 4. Первое поколение: разговор как с другом	15
Глава 5. Первое плюс: роль, контекст, задача	16
Глава 6. Второе поколение: личный агент	18
Глава 7. Второе плюс: инструменты, MCP, API, CLI	19
Глава 8. Третье поколение: workflow	20
Часть III. Дисциплина внедрения	22
Глава 9. Проверочные циклы: дисциплина вместо надежды	23
Глава 10. Третье плюс: многоагентность без восторженной глупости	24
Глава 11. Четвёртое поколение: ИИ как операционный слой бизнеса	25
Глава 12. Четвёртое плюс: память, правила, база знаний	26
Глава 13. Пятое поколение: ИИ как интерфейс ко всему	27
Глава 14. Пятое плюс: персональная операционная система	28
Глава 15. Шестое поколение: самооптимизирующиеся процессы	29
Глава 16. Шестое плюс: закрытые бизнес-контур	30
Глава 17. Седьмое поколение: агенты договариваются с агентами	31

Глава 18. Восьмое поколение: ИИ-native компании	32
Глава 19. Девятое и десятое поколения: симбиоз и инфраструктура	33
Часть IV. Где ИИ уже меняет рынок	35
Глава 20. Контент-заводы, видео, аватары и тренды	36
Глава 21. Как разработчику продавать ИИ-трансформацию хорошо	41
Глава 22. Как понять, что консультант понимает цифровую трансформацию	49
Часть V. Практические инструменты	52
Инструмент 1. Матрица первого внедрения	53
Инструмент 2. Что не надо автоматизировать первым	55
Инструмент 3. Форматы работы	56
Инструмент 4. 15 вопросов перед внедрением ИИ	57
Инструмент 5. Карта зрелости: как выбирать и что делать дальше	58
Послесловие. С чего на самом деле начинается трансформация	60
Если вы хотите применить это к своему бизнесу	60
Финальная формула	62

Об авторе

Константин Потапов — Senior Python Backend / R&D Engineer и консультант по цифровой трансформации. 16 лет в разработке и R&D: backend-системы, интеграции, базы данных, внутренние инструменты, CI/CD, Linux-инфраструктура, агентные workflow и прототипы цифровых продуктов.

Его рабочая позиция проста: не продавать «магический ИИ», а собирать управляемые цифровые контуры. Найти ручную петлю. Понять данные. Выбрать безопасный первый шаг. Собрать прототип. Встроить проверки. Довести до результата, который можно использовать после демонстрации.

Зона силы — перевод между бизнесом и инженерией. С одной стороны: скорость, заявки, отчёты, контроль, стоимость ошибки. С другой: API, базы данных, интеграции, агенты, права доступа, тесты, поддержка.

Сайт: potapov.me

Как написана эта книга

Эта книга не притворяется, что ИИ стоял в стороне.

Я использовал ИИ как рабочий инструмент: для сборки структуры, проверки повторов, редакторских проходов, подготовки схем, вёрстки и технической сборки PDF. Но рамка, позиция и ответственность здесь человеческие. ИИ не выбирал, что считать важным. Он не знает мою практику лучше меня. Он ускорял черновую работу, помогал видеть слабые места и собирать материал в форму.

Это ровно тот подход, о котором книга: не заменить человека машиной, а собрать контур, где человек задаёт смысл, критерии и границы, а ИИ берёт на себя часть тяжёлой работы.

Если коротко: книга написана не «вручную против ИИ» и не «ИИ вместо автора». Она сделана в рабочей связке. Так, как сегодня и стоит делать сложные экспертные материалы.

Сцена, с которой всё начинается

Представьте две компании в одной нише. У них похожие продукты, похожий рынок, похожие отделы продаж и маркетинга. Обе услышали про ИИ. Обе решили: пора попробовать.

Первая компания купила доступ к нейросетям. Сотрудники пишут посты, заголовки, иногда письма. На совещаниях звучит: «мы уже используем ИИ». Это правда. Но заявки обрабатываются так же медленно, отчёты собираются вручную, знания лежат в переписках, руководитель узнаёт о проблемах поздно. ИИ есть. Трансформации нет.

Вторая компания начала скучнее — и поэтому выиграла. Она выбрала один контур: входящие заявки. Разобрала, откуда они приходят, кто их классифицирует, где теряются данные, как быстро отвечает менеджер, какие статусы видит руководитель. Потом собрала агентный сценарий: первичная классификация, обогащение данных, черновик ответа, задача ответственному, ежедневная сводка рисков. Сначала всё подтверждал человек. Потом часть действий стала автоматической.

Через полгода первая компания всё ещё «экспериментировала с нейросетями». Вторая быстрее отвечала клиентам, видела просадки раньше и знала, какой следующий процесс автоматизировать.

Разница была не в модели. Разница была в дисциплине.

Одна компания использовала ИИ как игрушку. Другая превратила его в контур.

Предисловие

Эта книга написана для людей, которые принимают решения: владельцев бизнеса, руководителей, продактов, директоров по маркетингу, операционных менеджеров. Для тех, кому нужно не «поиграться с нейросетями», а понять, где ИИ уже можно встраивать в работу, где пока рано, сколько контроля оставлять человеку и как не купить красивую имитацию цифровой трансформации.

Но главный вопрос ещё острее: как не проспять технологический сдвиг и собрать с него дивиденды — деньгами, скоростью, качеством, управляемостью, карьерной позицией или долей рынка.

Для меня цифровая трансформация — не презентация про будущее. Это перевод конкретных процессов в управляемые цифровые контуры: с данными, инструментами, ответственностью, проверками и измеримым результатом. ИИ сейчас просто сделал этот разговор острее. Он показал, где у компании есть процесс, а где только привычка, Excel и героизм отдельных людей.

Разговоры об искусственном интеллекте часто начинаются не с того конца. Люди спрашивают: какая модель умнее? GPT или Claude? Gemini или локальная модель? Какая версия лучше пишет тексты, какая лучше кодит, какая меньше врёт.

Вопрос не бесполезный, но он уже не главный.

Модель сама по себе похожа на сильную лошадь без упряжки. Мощность есть. Направления нет. Можно восхищаться мускулами, спорить о породе и скорости, но поле она не вспашет, пока вокруг неё не появится обвязка: поводья, плуг, человек, маршрут, правила, остановка, проверка результата.

С ИИ происходит то же самое.

Правило: модель отвечает. Контур работает. Разница между ними — в инструментах, проверках и ответственности.

Первый уровень — спросить у модели что-нибудь в чате. Второй — дать ей роль и контекст. Третий — подключить инструменты. Потом появляются агенты, планы, память, проверки, внешние системы, права доступа, циклы качества, субагенты, рабочие процессы. В какой-то момент ИИ перестаёт быть “умным собеседником” и становится частью операционной среды.

Это не книга про магию и не сборник “100 промптов, которые изменят жизнь”. Промпты важны, но это первый этаж. Если остановиться на нём, быстро начинается самообман: вроде бы работа идёт, текстов много, идей много, планов много, а в реальности ничего не внедрено.

Здесь будет другая лестница. Практическая. От свободного чатбота до ИИ как инфраструктуры исполнения намерений.

И сразу честно: чем выше уровень, тем меньше ценность красивого запроса и тем больше ценность дисциплины. Контекст, доступы, проверки, регламенты, права, ограничения, память, метрики. Всё скучное. Всё то, что обычно хочется пропустить.

Пропустишь — получишь не агента, а болтливого стажёра с доступом к кнопкам.

Как читать эту книгу

Это не справочник по моделям и не обзор модных инструментов. Инструменты меняются слишком быстро. Сегодня один лидер, завтра другой, послезавтра у всех одинаковые функции. Гораздо устойчивее другой вопрос: как устроить внедрение так, чтобы оно приносило пользу бизнесу, а не производило видимость прогресса.

Я смотрю на ИИ как инженер и консультант по цифровым системам. Меня интересует не восторг от технологии, а контур: вход, действие, проверка, ответственность, результат. Где данные? Кто принимает решение? Что можно автоматизировать? Что нельзя отдавать машине? Как проверить, что агент не срезал углы? Как превратить разовую удачу в повторяемый процесс?

Если вам нужно выбрать подрядчика или консультанта для цифровой трансформации, эта книга должна помочь увидеть разницу между двумя типами людей. Одни продают впечатление: «подключим ИИ, будет современно». Другие начинают с процессов, ограничений, метрик и цены ошибки. Первые обычно делают демо. Вторые строят систему, которую можно использовать после демо.

Здесь я буду говорить именно о втором подходе.

Часть I. Зачем ловить сдвиг

Эта часть отвечает на главный вопрос читателя: где в цифровой трансформации появляются деньги, скорость и позиция — и как заметить это раньше рынка.

Глава 1. Как предугадывать будущее без магического шара

Будущее нельзя предсказать точно. Любой, кто говорит обратное, продаёт не экспертизу, а уверенность. Но направление сдвига можно увидеть раньше большинства. Для бизнеса этого достаточно. Деньги редко лежат в точном пророчестве. Они лежат в раннем распознавании: что уже стало возможным, что дешевет, что скоро станет стандартом и где конкуренты ещё спят.

Технологический сдвиг почти всегда проходит один и тот же путь. Сначала появляется дорогая игрушка для энтузиастов. Потом инструмент для тех, кто готов терпеть шероховатости. Потом преимущество для ранних внедренцев. Потом норма рынка. После этого дивиденды исчезают: то, что вчера давало отрыв, сегодня становится гигиеной.

ИИ сейчас проходит именно этот путь. В одних компаниях он всё ещё игрушка для текстов. В других — инструмент ускорения. В третьих — уже слой процессов, продаж, поддержки, разработки и управленческой видимости. Разница между этими состояниями и есть окно возможностей.

Где возникают дивиденды

Дивиденды цифровой трансформации — это не только прибыль. Прибыль часто приходит позже. Сначала появляются другие формы выигрыша.

Дивиденд скорости. Компания быстрее отвечает клиентам, быстрее готовит КП, быстрее проверяет гипотезы, быстрее выпускает материалы. Скорость превращается в обучение. Кто быстрее учится, тот раньше видит работающие ходы.

Дивиденд качества. Хороший ИИ-контур снижает разброс. Не один сильный сотрудник сделал хорошо, а процесс стабильно выдаёт приемлемый уровень и проходит проверку.

Дивиденд внимания. Руководитель перестаёт собирать картину вручную. Сводки, риски, отклонения и незакрытые петли поднимаются вверх автоматически.

Дивиденд масштаба. Маленькая команда начинает делать объём, который раньше требовал отдела. Не потому что люди стали сверхлюдьми, а потому что часть рутины стала машинной.

Дивиденд знания. Документы, переписки, решения, регламенты и заметки перестают лежать мёртвым архивом. Они становятся рабочей памятью организации.

Дивиденд контроля. Парадоксально, но правильно внедрённый ИИ не уменьшает контроль, а усиливает его: появляются логи, чеклисты, статусы, критерии, точки подтверждения.

Дивиденд позиции. Самый дорогой. Компания или специалист, которые раньше других поняли новый стандарт работы, занимают место до того, как оно стало очевидным для всех.

Как смотреть на сдвиг

Нужно смотреть не на новости, а на падение стоимости действия. Если действие резко дешевеет, рынок перестраивается. Подготовить черновик письма, собрать анализ конкурентов, проверить сайт, написать тесты, обработать входящие, сделать первичный отчёт — всё это дешевеет. Значит, старые процессы будут пересобраны.

Второй признак — повторяемость. Всё, что повторяется, имеет данные и понятный критерий качества, рано или поздно станет кандидатом на автоматизацию. Не обязательно полную. Часто сначала это будет режим: машина готовит, человек подтверждает.

Третий признак — переход от демо к контуру. Демо показывает, что технология возможна. Контур показывает, что она приносит бизнес-ценность. Именно здесь обычно нужен консультант: не чтобы восхититься возможностью, а чтобы довести её до процесса, прав, проверок и результата.

Поэтому поколения ИИ в этой книге — не футуристическая классификация. Это карта раннего обнаружения дивидендов. Если рынок вокруг вас сидит на первом поколении, а вы строите второе плюс, у вас есть окно преимущества. Если конкуренты генерируют тексты, а вы автоматизируете обработку заявок, вы играете в другой лиге.

Игрушка, инструмент, преимущество, гигиена

Технологии входят в бизнес четырьмя стадиями.

Игрушка. Люди пробуют, удивляются, показывают коллегам. Здесь много разговоров и мало ответственности. Польза случайная.

Инструмент. Отдельные специалисты начинают получать ускорение: быстрее писать, анализировать, искать, собирать черновики. Компания ещё не изменилась, но сильные сотрудники уже работают иначе.

Преимущество. Технология встраивается в процесс. Скорость, качество или видимость становятся лучше, чем у конкурентов. Здесь появляются дивиденды.

Гигиена. Так начинают работать все. Преимущество исчезает. То, что вчера давало отрыв, сегодня стало минимальным стандартом.

Главный вопрос руководителя: на какой стадии находится технология именно в вашем рынке? Если для конкурентов ИИ ещё игрушка, а вы можете превратить его в инструмент, у вас есть локальное преимущество. Если у сильных игроков это уже процесс, а у вас только эксперименты, вы не исследуете будущее. Вы отстаёте.

Перед тем как идти по поколениям, держите в голове простую оптику: каждое поколение отвечает не на вопрос «что умеет модель», а на вопрос «какой новый дивиденд может получить бизнес».



Часть II. От промпта к рабочему контуру

Здесь начинается лестница зрелости: от разговора с моделью до агентных процессов, которые меняют работу компании.

Лестница состоит из одиннадцати ступеней — от нулевой до десятой:

- **0:** просто модель — генерация текста в чате.
- **1:** свободный разговор, собеседник.
- **1+:** роль, контекст, задача, формат.
- **2:** личный агент с доступом к файлам и командам.
- **2+:** инструменты, MCP, API, внешние системы.
- **3:** workflow — повторяемый автоматизированный процесс.
- **3+:** многоагентность, параллельные роли.
- **4:** ИИ как операционный слой бизнеса.
- **4+:** память, правила, корпоративная база знаний.
- **5:** ИИ как единый интерфейс ко всем системам.
- **5+:** персональная операционная система.
- **6:** самооптимизирующиеся процессы по метрикам.
- **6+:** закрытые бизнес-контуры с ограниченной автономией.
- **7:** агенты договариваются с агентами других компаний.
- **8:** ИИ-native компания — люди управляют контурами.
- **9–10:** когнитивный слой и невидимая инфраструктура.

Каждая ступень отвечает не на вопрос «что умеет модель», а на вопрос «какой новый дивиденд получает бизнес».

Глава 2. Почему “хороший промпт” больше не главное

В начале всем казалось, что секрет в формулировке. Надо правильно попросить. Написать “действуй как эксперт”, добавить “думай шаг за шагом”, попросить “сначала составь план”. Иногда это правда помогало. Особенно на ранних моделях, где контекст был коротким, а поведение нестабильным.

Но промптинг — это не внедрение. Это разговор.

Промпт говорит модели, что мы от неё хотим. Контекст даёт ей материал. Харнес строит среду, в которой модель может довести работу до результата.

Харнес — неудобное слово, но точное. Это вся обвязка вокруг модели: память, инструменты, файлы, правила, проверки, циклы, права, доступы, планы, субагенты, журналы действий. Именно харнес отличает чатик от агента.

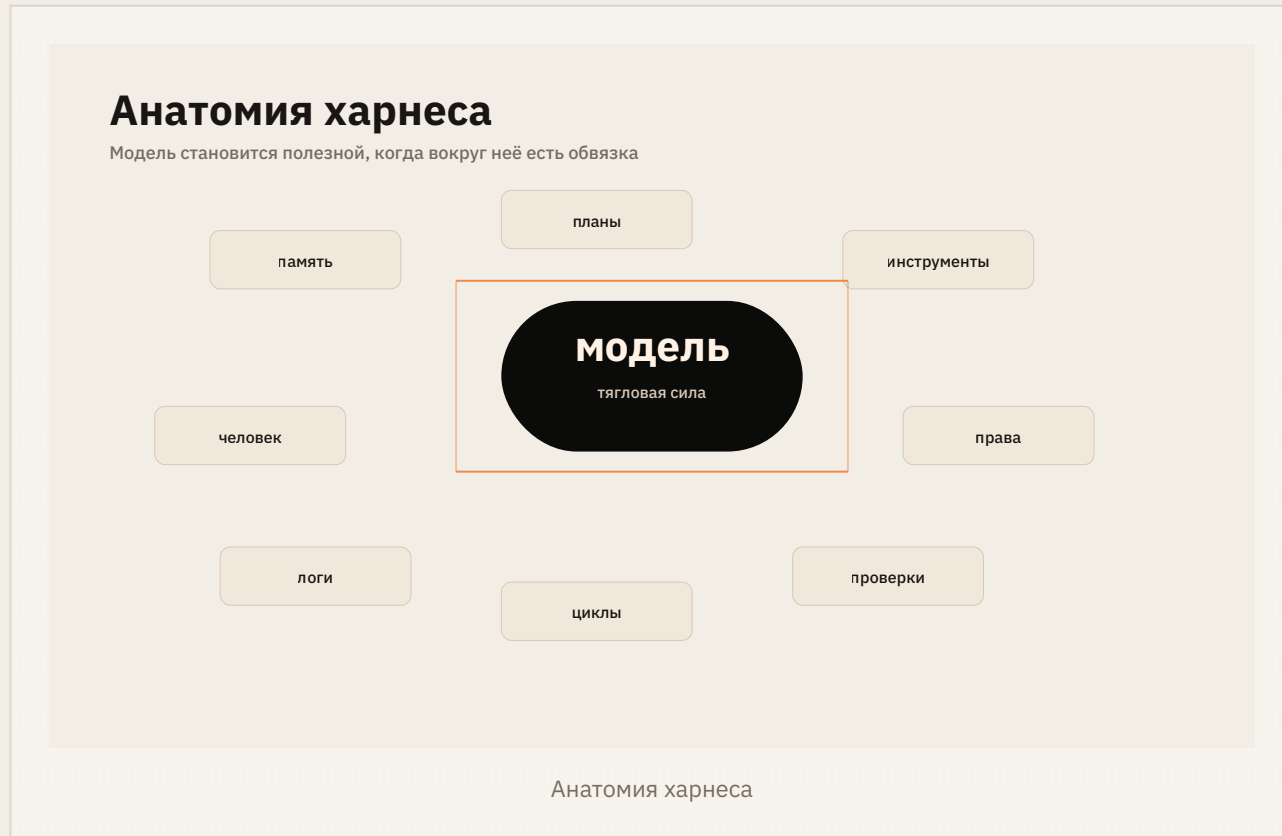
В обычном чате модель может написать: “вот как исправить ошибку”. В агентной среде она открывает проект, читает файлы, меняет код, запускает тесты, видит ошибку, исправляет снова, оставляет отчёт. Модель может быть одна и та же. Разница — в обвязке.

Отсюда главный сдвиг мышления:

Если агент ошибся, не надо первым делом переписывать промпт. Надо спросить: что в среде позволило этой ошибке случиться?

Забыл стиль проекта — нужен файл правил. Потерял цель — нужен план и критерии приёмки. Сказал “готово”, но не проверил — нужен обязательный шаг верификации. Сломал формат — нужна схема. Начал галлюцинировать — нужны источники и ограничения. Повторил ошибку второй раз — это уже не ошибка модели, а дырка в процессе.

Так появляется инженерия харнеса.



Три эпохи

Сначала был промптинг. Мы учили себя говорить с моделью.

Потом пришла инженерия контекста. Мы начали давать модели документы, базы знаний, репозитории, переписки, свежие данные, инструменты, RAG (поиск по базе знаний) и MCP (протокол подключения внешних инструментов к модели). Это был мощный шаг: модель перестала “помнить всё сама” и научилась брать нужное из окружения.

Но у большого контекста есть тёмная сторона. Его легко перегрузить. Чем больше в диалоге мусора, тем выше риск, что агент начнёт путать факты, забывать ограничения и считать невыполненные задачи выполненными. Большое окно — не бесплатная мудрость. Иногда это просто большая свалка.

Третья эпоха — харнес. Здесь вопрос уже не в том, сколько текста сунуть в модель. Вопрос в том, как организовать работу: какие шаги, какие проверки, кто принимает решение, где остановка, где человек, где автоматический тест, где лог, где право на ошибку.

Коротко:

- промптинг учит модель отвечать;
- контекст учит её учитывать материал;
- харнес учит её работать.

Глава 3. Нулевая точка: просто модель

Нулевое поколение — это ИИ как генератор текста. Пользователь пишет: “объясни”, “переведи”, “придумай”, “напиши пост”. Модель отвечает. Иногда хорошо. Иногда уверенно врёт. Иногда попадает в тон. Иногда выдаёт аккуратную вату.

В этом нет проблемы. Уровень полезный. Он снимает страх белого листа и даёт быстрый черновик. Но здесь почти нет внедрения. Нет памяти. Нет проверки. Нет доступа к реальности. Нет ответственности за результат.

Рабочий способ использовать нулевой уровень — просить варианты, а не финальный ответ.

Например:

Дай десять названий. От банального к смелому. Потом выбери три и объясни, почему они сильнее.

Или:

Объясни идею тремя способами: для новичка, для предпринимателя и для инженера.

Слабый способ — верить первому ответу. Модель звучит уверенно по умолчанию. Это не признак истины. Это особенность интерфейса.

Практический критерий простой: нулевое поколение хорошо, если оно ускоряет начало работы. Плохо, если оно подменяет экспертизу.

Выигрыш: быстрый старт

Выигрыш здесь маленький, но быстрый: снять трение старта. Быстрее получить черновик, объяснение, список вариантов, перевод, структуру. Это экономит минуты и часы, но почти не создаёт стратегического преимущества. Такой уровень быстро копируется любым конкурентом.

Глава 4. Первое поколение: разговор как с другом

Первое поколение — свободный разговор. Человек приходит не за документом, а за собеседником. “Мне тревожно”. “Помоги подумать”. “Я застрял”. “Давай обсудим идею”.

Это важный слой, потому что многие реальные задачи начинаются не с ТЗ, а с каши в голове. Человек не всегда знает, чего хочет. Иногда ему нужно сначала услышать себя.

Но здесь же первая ловушка. Разговор создаёт ощущение движения. Можно час обсуждать проект и не сделать ни одного действия. Можно получить поддержку и остаться в той же точке. Можно принять мягкое согласие модели за подтверждение своей правоты.

Правильный запрос на этом уровне звучит не так:

Поддержи меня.

А так:

Разложи мой рассказ на факты, эмоции, гипотезы и действия. Не соглашайся автоматически. Покажи, где я могу себя обманывать. В конце дай один следующий шаг.

Первое поколение становится полезным, когда ИИ не убаюкивает, а помогает отделить реальность от тумана.

Хороший результат разговора — не “мне стало приятно”. Хороший результат — один честный следующий шаг.

Выигрыш: ясность

Выигрыш — ясность. Руководитель или специалист быстрее распутывает хаос, отделяет факты от эмоций и принимает первое решение. Это ещё не цифровая трансформация, но уже снижение внутреннего шума. Особенно ценно для одиночных предпринимателей и маленьких команд, где некому быстро отзеркалить мысль.

Глава 5. Первое плюс: роль, контекст, задача

Здесь начинается прикладная работа. Пользователь уже не просто разговаривает. Он задаёт роль, аудиторию, цель, ограничения и формат.

Например:

Ты эксперт по рекламе. Аудитория — владельцы малого бизнеса, которые льют трафик на лендинг и не получают заявок. Продукт — аудит первого экрана. Нужно 10 креативов. Тон уверенный, без инфоцыганства. Формат: заголовок, текст, визуальная идея.

Это уже намного лучше свободного “придумай рекламу”. Модель получает рамку. Результат становится применимым.

Но есть важное “но”. Для сложных задач нельзя сразу просить финальный ответ. Модель почти всегда попытается срезать углы. Она даст стратегию без исследования, лендинг без понимания аудитории, план без критериев, тесты без связи с материалом.

Поэтому появляется первый серьёзный приём:

Сначала составь план того, как ты будешь решать задачу. Какие данные нужны? Какие этапы? Какие риски? По каким критериям проверим результат?

Для трудных задач ещё лучше:

Составь план для плана. Как правильно спроектировать работу над этой задачей?

Это звучит избыточно, пока не попробуешь. Потом становится ясно: хороший агент ленится не потому, что “плохой”. Он ленится потому, что ему разрешили перепрыгнуть через мышление.

Выигрыш: качество мышления

Выигрыш — компания быстрее производит не просто текст, а варианты офферов, гипотез, писем, сценариев, исследований. В маркетинге и продажах это уже может давать деньги: больше проверенных формулировок, быстрее упаковка предложений, меньше зависимость от пустого листа. Но преимущество держится только там, где есть хороший контекст и критерии.

Глава 6. Второе поколение: личный агент

Второе поколение начинается там, где ИИ получает руки.

Он читает файлы. Запускает команды. Работает с браузером. Правит код. Создаёт документы. Проверяет результат. Может сделать цепочку действий, а не просто написать совет.

Запрос меняется:

Проверь проект, найди ошибку, исправь, запусти тесты и дай отчёт.

Это уже не “как бы ты сделал”. Это “сделай”.

Здесь план становится не украшением, а поводом. Если задача не однострочная, сначала нужен план. Не потому что человек обязан читать триста строк markdown. План нужен агенту. Он заставляет его понять задачу, описать подход, зафиксировать критерии готовности и не объявить победу слишком рано.

Человеку достаточно спросить:

Дай TLDR плана. Где главный риск? Какие решения ты принял? Что мне нужно утвердить?

Правило: план — это не бюрократия. Это защита от распада контекста.

План — это контрольная точка. Если сессия оборвалась, можно начать новую и продолжить по плану. Если агент потерял контекст, план возвращает его на рельсы. Если результат спорный, план показывает, где он свернул.

На этом уровне полезно держать жёсткое правило:

Правило: первый черновик агента не считается готовым, пока он не прошёл проверку.

Не “агент сказал готово”. А тесты прошли, файл создан, документ открылся, чеклист закрыт, ошибки названы.

Выигрыш: исполнительская сила

Выигрыш — перенос части исполнительской работы с человека на агента. Это уже экономия часов, а иногда и дней: правки в коде, подготовка документов, разбор файлов, локальные проверки, сбор отчётов. Для консультанта или небольшой команды это особенно ценно: можно продавать не “советы”, а быстрее производить артефакты.

Глава 7. Второе плюс: инструменты, MCP, API, CLI

Агент с инструментами — это уже другой класс системы. Он может работать с GitHub, CRM, базой знаний, календарём, почтой, документами, браузером, внутренними API (программными интерфейсами систем).

Но здесь люди часто делают глупость: подключают всё подряд. Просто потому что можно.

Это плохой путь. Каждый инструмент имеет цену: контекст, безопасность, сложность, права, риск ошибки. У агента должны быть не все возможные руки, а ровно те, которые нужны для процесса.

MCP (протокол стандартного подключения внешних инструментов) удобен, когда нужна типизированная интеграция, авторизация, устойчивый протокол. CLI (интерфейс командной строки) или простой скрипт часто лучше, когда достаточно одной команды и не хочется забивать контекст описаниями десятков инструментов.

Практическое правило:

Не подключай инструмент “на всякий случай”. Подключай под повторяемый процесс.

Например, если агент должен раз в неделю собирать лиды, проверять сайты и готовить отчёт, ему нужны конкретные доступы: где брать лиды, как открывать сайты, куда класть отчёт, какие поля заполнять, чего не трогать.

Особенно осторожно с правами. Сначала агент читает и предлагает. Потом готовит черновики. Потом действует после подтверждения. И только когда контур стабилен, можно давать малую автономию.

Свобода без границ — не автономность. Это авария с красивым интерфейсом.

Выигрыш: связность систем

Выигрыш — связность систем. Агент перестаёт быть отдельным окном и начинает двигать данные между реальными системами: CRM, базой знаний, документами, задачами, репозиториями. Деньги появляются там, где раньше человек был переносчиком информации. Если он копировал, сверял, классифицировал и создавал однотипные задачи — это кандидат на быстрый выигрыш.

Глава 8. Третье поколение: workflow

Третье поколение — это когда ИИ закрывает не одну задачу, а повторяемый процесс.

Не “напиши письмо”, а:

Каждый день проверь новые заявки, классифицируй их, подготовь черновики ответов и пришли сводку.

Не “сделай анализ конкурентов”, а:

Раз в неделю собери изменения у конкурентов, выдели новые офферы, цены, посадочные страницы и предложи, что нам проверить.

Здесь главный документ — регламент. Даже короткий. В нём должны быть входы, выходы, шаги, ограничения, критерии качества и случаи, когда агент обязан остановиться.

Если процесса нет, автоматизировать нечего. Можно только ускорить хаос.

Хороший запрос на этом уровне:

Составь регламент процесса. Укажи триггер, входные данные, шаги, результат, критерии качества, ошибки и ситуации, когда нужно спросить человека.

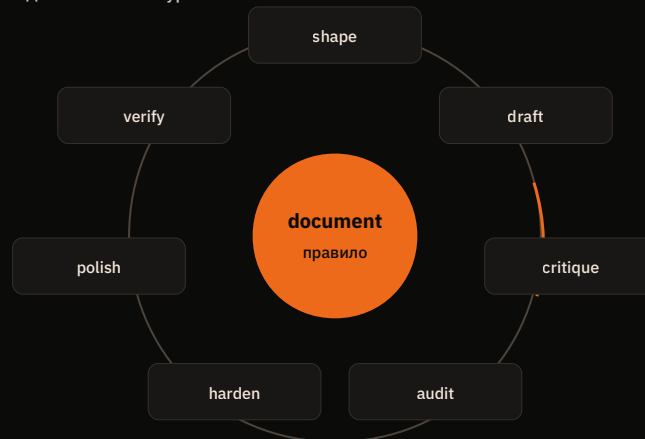
Workflow ценен предсказуемостью. Один раз хорошо собранный процесс можно повторять. Потом улучшать. Потом превратить в skill, скрипт, cron-задачу или внутреннюю агентную связку.

Выигрыш: регулярная экономия

Выигрыш — регулярная экономия, а не разовая удача. Если процесс запускается каждый день или каждую неделю, эффект накапливается: быстрее обработка заявок, стабильнее отчёты, меньше забытых действий, выше управляемость. Здесь ИИ впервые начинает работать как операционный актив.

Проверочный цикл

Не один промпт, а производственный контур качества



Проверочный цикл

Часть III. Дисциплина внедрения

ИИ полезен ровно настолько, насколько хорошо вокруг него собрана рабочая среда: планы, проверки, ограничения, метрики и человеческое решение.

Эта часть начинается с инструмента, который нужен на любом уровне лестницы — проверочные циклы. А затем продолжает подъём: от многоагентности до ИИ-native компаний и инфраструктуры.

Глава 9. Проверочные циклы: дисциплина вместо надежды

Самый опасный способ работать с ИИ — верить, что сильная модель с первого раза сделает “нормально”. Иногда делает. Часто нет. И почти всегда будет звучать увереннее, чем заслуживает результат.

Поэтому серьёзная работа строится не вокруг одного промпта, а вокруг проверочного цикла.

Минимальный цикл:

1. задать форму;
2. сделать черновик;
3. раскритиковать;
4. проверить по чеклисту;
5. укрепить крайние случаи;
6. отполировать;
7. подтвердить соответствие критериям;
8. зафиксировать новое правило.

Я называю это *named passes* (именованные проходы): не абстрактное “улучши”, а конкретный проход с конкретной целью.

Shape (Рамка) — задать рамку. Кто читатель, какая цель, какой формат, что запрещено, что считается хорошим результатом.

Draft (Черновик) — сделать первую версию. Не финальную. Просто материал, с которым можно работать.

Critique (Критика) — найти слабые места. Где мутно, где вода, где нарушена логика, где текст звучит как нейросеть, где нет практической пользы.

Audit (Аудит) — проверить по формальным критериям. Не нравится/не нравится, а проходит/не проходит.

Harden (Укрепление) — проверить крайние случаи. Что будет с новичком? Где можно угадать ответ? Где формулировка двусмысленна? Где процесс ломается на реальных данных?

Polish (Полировка) — улучшить язык и подачу. Только после проверки конструкции.

Verify (Проверка) — показать, какие критерии выполнены, какие риски остались.

Document (Документирование) — зафиксировать правило, если ошибка повторилась.

На этом принципе можно строить всё: тексты, тесты, образовательные материалы, код, коммерческие предложения, лендинги, PDF.

В Tech Path Finder — платформе для подготовки к техническим собеседованиям с более чем 7300 вопросами по 57 IT-направлениям — этот подход особенно важен. Учебный материал не готов, пока не проверено, чему он учит, какой навык проверяет тест, нет ли вопросов “на угадку”, совпадает ли тест с содержанием урока, не подменяет ли сложность понимание.

Это и есть взрослая работа с ИИ. Не “сгенерируй мне курс”. А “собери материал, проверь его как методист, проверь тесты как экзаменатор, найди ложную сложность, исправь, покажи, что критерии закрыты”.

Промптинг учит ИИ красиво отвечать. Харнес учит ИИ действовать. Проверочные циклы учат ИИ не халтурить.

Глава 10. Третье плюс: многоагентность без восторженной глупости

Многоагентные системы выглядят красиво. Один агент исследует, другой пишет, третий критикует, четвёртый проверяет, пятый собирает. Почти маленькая компания.

Иногда это правда полезно. Особенно когда нужны разные перспективы: стратегия, переговоры, исследование рынка, креативный отбор, аудит сложного решения.

Но много агентов — не бесплатное качество.

Если пять агентов каждый надёжен на 95%, цепочка не становится 95-процентной. Ошибки накапливаются. Общая надёжность падает. Чем длиннее цепочка, тем больше мест, где можно потерять смысл, неверно передать вывод, принять слабую гипотезу за факт.

Поэтому для большинства задач лучше один сильный агент с хорошими инструментами, планом и проверкой, чем пять болтливых персонажей.

Многоагентность нужна, когда роли действительно различаются:

- исследователь собирает факты;
- исполнитель делает черновик;
- критик ищет слабые места;
- судья выбирает лучший вариант;
- редактор собирает финальный артефакт.

Главное — не забывать финальную ответственность. Кто собирает? Кто проверяет? Кто говорит “готово”? Если этого нет, многоагентность превращается в хор.

Выигрыш: параллельность

Выигрыш — параллельность и разнообразие взглядов. Можно быстрее исследовать рынок, получить несколько стратегий, провести внутреннюю критику, сравнить подходы. Это особенно полезно в консалтинге, продуктовой работе и сложной упаковке. Но прибыль появляется только при сильном финальном синтезе. Без него многоагентность создаёт шум.

Глава 11. Четвёртое поколение: ИИ как операционный слой бизнеса

На четвёртом уровне ИИ встраивается в функции бизнеса: продажи, поддержку, маркетинг, разработку, HR, финансы, документооборот.

Здесь уже нельзя говорить “внедрим ИИ”. Это слишком расплывчато. Внедрять надо в конкретный контур.

Например: обработка входящей заявки.

Агент получает заявку, классифицирует её, ищет контекст, готовит черновик ответа, создаёт задачу, уведомляет ответственного, записывает результат. Всё. Узкий контур. Понятный вход. Понятный выход. Можно измерить скорость, ошибки, экономию времени.

Так и надо начинать.

Слабое внедрение начинается с презентации про трансформацию. Сильное — с одного процесса, где каждый день теряется время или качество.

На этом уровне особенно важны права. Агент поддержки может предлагать ответ, но не обещать скидку. Агент продаж может подготовить КП, но не менять цену без подтверждения. Агент аналитики может найти риск, но не переписать план продаж без человека.

ИИ в бизнесе должен быть не фокусником, а исполнительным слоем с понятными границами.

Выигрыш: меньше операционного трения

Выигрыш — снижение операционного трения. Быстрее заявки, меньше ручных стыков, прозрачнее статусы, ниже зависимость от памяти сотрудников. Для руководителя это уже не “нейросеть пишет тексты”, а способ уменьшить стоимость процесса и повысить скорость реакции бизнеса.

Глава 12. Четвёртое плюс: память, правила, база знаний

Если агент каждый раз работает как в первый день, это не внедрение. Это аренда умного незнакомца.

На уровне 4+ у агента появляется память: документы, прошлые решения, стиль компании, политика скидок, история клиентов, база знаний, регламенты, примеры хорошей работы.

Память не обязана быть мистической. Часто это просто аккуратная папка с markdown-файлами, Outline, Notion, Obsidian, Google Drive или репозиторий с правилами.

Важно другое: память должна быть пригодна для работы.

Если в неё сложить хаос, агент начнёт воспроизводить хаос. Если там старые решения, неотмеченные исключения и противоречивые инструкции, модель будет звучать уверенно и ошибаться системно.

Практическое правило:

Всё, что ты не хочешь объяснять второй раз, должно стать правилом, заметкой, регламентом, примером или навыком (skill).

И второе:

Всё устаревшее должно удаляться или помечаться как устаревшее.

Память усиливает не только знание. Она усиливает и мусор.

Выигрыш: память как актив

Выигрыш — превращение опыта в актив. Решения, регламенты, переписки, исследования и примеры перестают исчезать в головах и чатах. Новые сотрудники быстрее входят в контекст, старые решения легче проверять, агент отвечает не “вообще”, а в логике конкретной компании. Это снижает потери знания и зависимость от отдельных людей.

Глава 13. Пятое поколение: ИИ как интерфейс ко всему

На пятом уровне человек перестаёт ходить по десяткам приложений. Он говорит намерение, а ИИ работает с системами.

Покажи клиентов, которые требуют внимания.
Найди, почему просела конверсия.
Подготовь отчёт по проектам с риском срыва сроков.
Запланируй встречи с горячими лидами.

Это не просто удобство. Это смена интерфейса. Человек взаимодействует не с CRM, календарём, почтой и таблицей по отдельности, а с задачей.

Но здесь резко растёт цена ошибки. Если ИИ становится входной точкой ко всему, нужны ограничения: что можно читать, куда можно писать, что требует подтверждения, где нужен лог, как откатить действие.

Естественный язык не заменяет архитектуру прав. Он только делает её ещё нужнее.

Выигрыш: короче путь к решению

Выигрыш — сокращение управленческого расстояния. Руководителю не нужно обходить десять систем, чтобы понять состояние продаж, проектов или рисков. Он задаёт вопрос бизнесу напрямую. Это даёт выигрыш во внимании и скорости решений.

Глава 14. Пятое плюс: персональная операционная система

Персональный ИИ — это не просто помощник, который помнит имя и любимый стиль письма. Это слой, который знает проекты, цели, ограничения, финансовую реальность, привычные самообманы, рабочий ритм.

Такой ИИ может сказать неприятное:

Ты называешь это исследованием, но третий день избегаешь продаж.

Или:

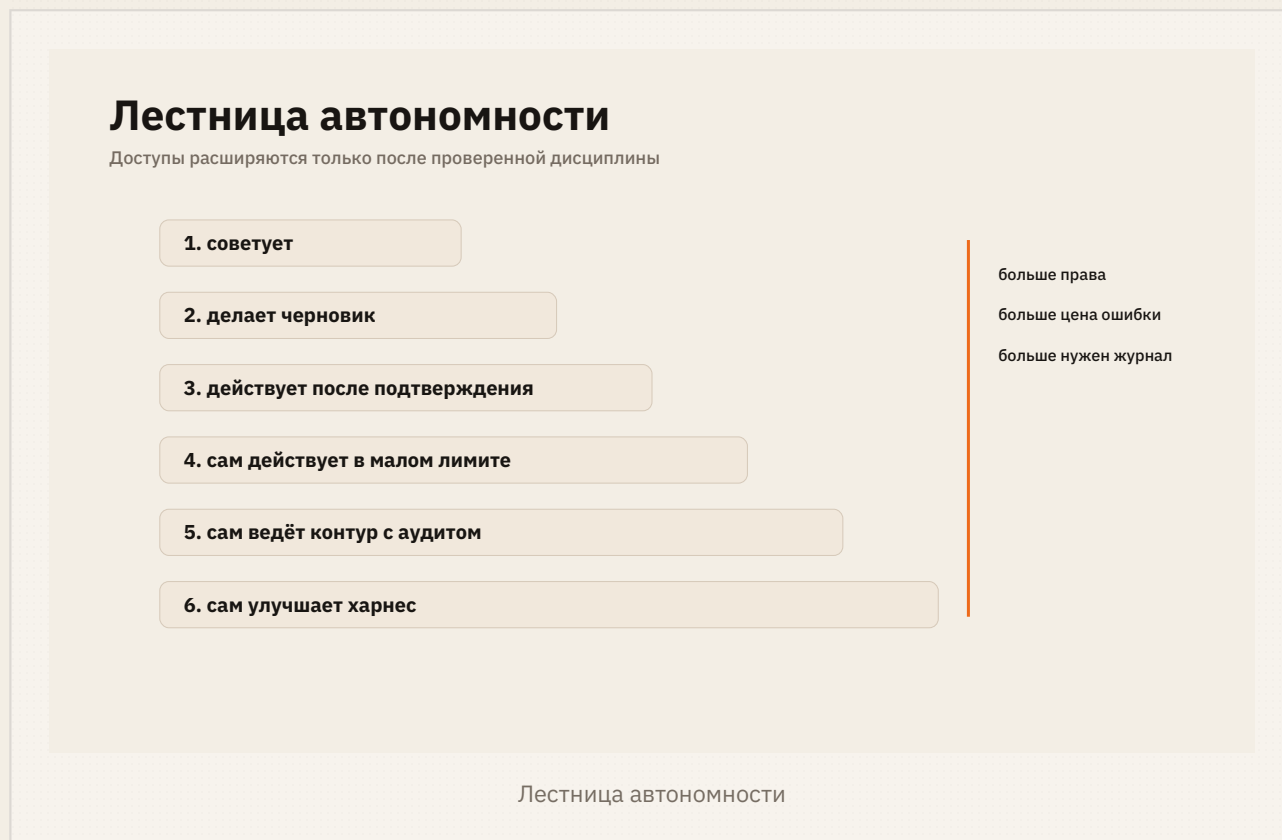
Из этих пяти задач деньги может принести только одна. Остальные создают ощущение занятости.

Это ценно. Но только если пользователь сам хочет честности, а не бесконечного одобрения.

Персональная ОС должна помогать не врать себе. Иначе она станет красивым зеркалом, которое всегда ставит хороший свет.

Выигрыш: личная собранность

Выигрыш — личная стратегическая собранность. Для предпринимателя, консультанта или руководителя это может быть дороже экономии времени: меньше имитации занятости, яснее приоритеты, быстрее выход к деньгам, клиентам и решениям. Такой ИИ усиливает не производительность вообще, а способность держать курс.



Глава 15. Шестое поколение: самооптимизирующиеся процессы

На шестом уровне ИИ не только выполняет процесс, но и улучшает его по метрикам.

Например: запуск рекламных гипотез. Агент предлагает идеи, готовит креативы, смотрит результат, отключает слабое, усиливает сильное, раз в неделю приносит выводы.

Звучит мощно. Но здесь главный вопрос — метрика.

Если выбрать лайки, система будет гнаться за лайками. Если выбрать количество текстов, будет производить тексты. Если выбрать скорость закрытия задач, может начать закрывать их плохо.

ИИ оптимизирует то, что вы ему дали. И если вы дали неправильную метрику, он станет очень полезным инструментом для движения не туда.

Нужен журнал гипотез:

- что проверяем;
- почему это может сработать;
- какой результат ждём;
- что получилось;
- что меняем дальше.

Самооптимизация без журнала быстро превращается в шаманство с графиками.

Выигрыш: быстрее учиться

Выигрыш — ускоренное обучение системы. Компания не просто делает больше, она быстрее понимает, что работает. В рекламе, продажах, поддержке и продуктовых экспериментах это напрямую связано с деньгами: меньше трат на слабые гипотезы, быстрее усиление сильных, короче путь от идеи до вывода.

Глава 16. Шестое плюс: закрытые бизнес-контур

Здесь агент получает право действовать в рамках бюджета, правил и целей.

Например:

В рамках 50 000 рублей в месяц приводи заявки на аудит лендингов. Не использовать кликбейт. CPL не выше заданного уровня. Всё выше лимита — на подтверждение.

Это уже серьёзная автономность. Вводить её надо слоями:

1. агент только анализирует;
2. готовит рекомендации;
3. делает черновики действий;
4. действует после подтверждения;
5. действует сам в малом лимите;
6. ведёт контур с аудитом.

Сразу дать агенту деньги, рекламу, клиентов и право действовать — не смелость. Это плохое управление.

Нормальная автономность выглядит скучно. У неё есть лимиты, журналы, stop-loss, отчёты, роли и право человека остановить систему.

Выигрыш: масштаб без раздувания команды

Выигрыш — масштабирование результата без пропорционального роста команды. Если контур безопасно ведёт рекламу, поддержку, мониторинг или первичную обработку сделок, бизнес получает рычаг: больше действий при том же управленческом внимании. Но это уже зона высокой цены ошибки, поэтому прибыль покупается дисциплиной.

Глава 17. Седьмое поколение: агенты договариваются с агентами

На уровнях 7 и выше «плюс»-варианты не выделяются отдельно: технологии здесь ещё достаточно экспериментальны, и чёткой границы между базовым уровнем и его зрелой версией пока не сложилось. Поэтому главы идут без пары.

Следующий слой — межорганизационные агенты. Агент клиента общается с агентом подрядчика. Агент юриста проверяет договор. Агент бухгалтера сверяет оплату. Агент проекта обновляет сроки.

Это экономит огромное количество переписки. Но принесёт новый вопрос: кто имеет право говорить от имени человека?

Агент должен различать черновик, намерение, обещание и юридически значимое действие. Без этого рынок утонет в “мой агент вроде договорился”.

Нужны цифровые доверенности, лимиты, проверка полномочий, журналы переговоров, подтверждение важных условий.

Иначе автоматизация коммуникации станет автоматизацией недоразумений.

Выигрыш: быстрее договариваться

Выигрыш — скорость сделок и согласований. Поиск подрядчиков, сверка условий, подготовка документов, первичные переговоры, закупки и бронирования могут занимать меньше человеческого времени. Кто раньше встроится в такие протоколы доверия, тот быстрее будет находить партнёров и закрывать рутинные согласования.

Глава 18. Восьмое поколение: ИИ-native компании

ИИ-native компания — это не компания без людей. Это компания, где люди меньше делают руками и больше управляют контурами.

Маленькая команда может вести маркетинг, продажи, поддержку, аналитику, документы и продуктовую работу с плотностью, которая раньше требовала отделов.

Но здесь снова нужна дисциплина. Всё, что повторяется больше двух раз, превращается в workflow, skill, регламент, шаблон, CLI или проверочный цикл.

Человек остаётся там, где нужны вкус, стратегия, ответственность, переговоры, этика и право сказать “нет”.

Люди рулят. Агенты выполняют.

Красивая фраза. Но она работает только при одном условии: люди действительно рулят, а не просто смотрят, как агенты производят горы артефактов.

Выигрыш: новая экономика маленьких команд

Выигрыш — новая экономика маленьких команд. Небольшая студия, агентство или продуктовая команда может конкурировать с более тяжёлыми структурами за счёт скорости, плотности производства и накопленных агентных регламентов. Это уже не экономия часов. Это изменение конкурентной позиции.

Глава 19. Девятое и десятое поколения: симбиоз и инфраструктура

Девятое поколение — ИИ как когнитивный слой рядом с человеком. Он помогает видеть паттерны, помнить обещания, отделять движение от имитации, держать фокус.

Это тонкая территория. Здесь легко перейти грань между помощью и зависимостью. Агентная работа затягивает. Она похожа на игру, где всегда можно построить ещё один инструмент, улучшить ещё один процесс, запустить ещё один проект.

Главный вопрос остаётся неприятным:

Это кому-нибудь нужно?

Если да — идём к рынку. Если нет — честно называем это личным инструментом или экспериментом. В этом нет стыда. Стыд начинается там, где стройка выдается за бизнес.

Десятое поколение — ИИ как невидимая инфраструктура реальности. Человек не открывает приложение и не выбирает инструмент. Он формулирует намерение: запустить продукт, организовать поездку, собрать команду, провести исследование, восстановить здоровье процесса.

Система сама собирает план, людей, документы, риски, бюджет, коммуникации и контрольные точки.

Звучит как финал. На самом деле это возвращает нас к началу: кто задаёт цели? Кто ставит ограничения? Кто имеет право остановить систему? Где граница между помощью и управлением человеком?

Чем мощнее ИИ, тем важнее человеческое “нет”.

Выигрыш: позиция в будущем стандарте

Выигрыш — позиция в будущем стандарте работы. На девятом уровне выигрывает человек, который думает и действует вместе с ИИ быстрее, честнее и собраннее. На десятом выигрывают организации, которые заранее построили инфраструктуру намерений: данные, права, память, процессы, доверие. Когда это станет нормой, догонять будет дорого.



Часть IV. Где ИИ уже меняет рынок

Контент, продажи, данные и консалтинг — зоны, где дивиденды видны быстрее всего. Но только если не путать производство шума с управляемым контуром.

Глава 20. Контент-заводы, видео, аватары и тренды

Про контент-заводы легко говорить красиво. «Берём тренды, генерируем сценарии, делаем видео, публикуем во все каналы, собираем метрики». На схеме это выглядит почти безупречно.

В жизни первый вопрос грубее: сколько стоит одна сырая минута видео, сколько времени съедает обход ограничений и сколько мусора нужно выбросить, прежде чем появится материал, который не стыдно подписать своим именем.

Я пробовал собирать такие контуры сам. Не как сторонний наблюдатель, а руками: тексты, сценарии, картинки, видео, озвучка, аватары, автоматическая переупаковка. И довольно быстро понял неприятную вещь: ИИ резко удешевляет отдельные операции, но весь контур целиком всё равно требует дисциплины, инфраструктуры и денег.

С чего начинается реальный контент-завод

Не с кнопки «сгенерировать пост».

Он начинается с вопроса: какой сигнал рынка мы хотим поймать и во что его превратить? В статью? В короткое видео? В карточку? В письмо? В рекламный креатив? В кусок лендинга? В гипотезу для оффера?

Если этого вопроса нет, завод быстро превращается в генератор правильного шума. Материалы вроде есть. Публикации вроде идут. Но за ними нет угла зрения, проверки и следующего шага.

Хороший контент-завод делает не «много контента». Он делает короткий цикл:

сигнал → гипотеза → материал → публикация → реакция → вывод → следующая версия.

Именно цикл важен. Не количество постов.

Мой маленький AginX

В какой-то момент я собрал свой небольшой самописный контент-завод, рабочее название — AginX. Это не был «продукт на рынок» с красивой упаковкой. Скорее лаборатория: проверить, можно ли на старших моделях собрать контур, который помогает не просто писать тексты, а двигать материал между форматами.

Что я хотел получить:

- брать исходную мысль или тему;
- разворачивать её в несколько углов;
- превращать в структуру статьи;
- вытаскивать из неё сценарий видео;
- готовить короткие версии для соцсетей;
- подбирать визуальные направления;
- сохранять стиль и позицию, а не каждый раз начинать с пустого промпта.

На бумаге это звучит просто. На практике быстро появляются швы.

Модель может написать аккуратный текст, но часто сглаживает голос. Может предложить десять заголовков, но половина будет похожа на заголовки из чужого корпоративного блога. Может сделать структуру ролика, но не понимает, где зритель устанет. Может придумать визуал, но не знает, сколько будет стоить довести его до нормального видео.

Поэтому AginX стал для меня не доказательством «ИИ всё сделает», а доказательством обратного: без авторского отбора, правил, памяти, проверки и стоимости инфраструктуры контент-завод быстро производит не рычаг, а новую форму рутины.

Российская реальность: сервисы доступны не всем

В западных демонстрациях контент-завод часто выглядит так: подключили ElevenLabs, Runway, HeyGen, Midjourney, пару API, CRM, планировщик публикаций — и всё поехало.

Из России эта картинка ломается.

Часть сервисов недоступна напрямую. К 11Labs и ряду других инструментов нельзя просто прийти, спокойно оплатить и встроить в рабочий процесс. Где-то мешает география, где-то платежи, где-то аккаунты, где-то риск внезапной блокировки. Даже если технически обход находится, он добавляет трение: прокси, зарубежные карты, посредники, нестабильные доступы, ручные костыли.

Это не политическая жалоба. Это производственная себестоимость.

Если из-за обхода блокировок сырая минута FullHD-видео обходится в 200+ рублей ещё до нормальной редакторской доводки, «дешёвое производство контента» перестаёт быть таким дешёвым. Особенно если половину вариантов всё равно придётся выбросить.

И вот здесь начинается взрослая экономика контент-завода. Считать нужно не цену генерации одного файла, а цену принятого материала:

- сколько стоит черновик;
- сколько вариантов ушло в мусор;
- сколько времени заняла проверка;
- сколько стоила озвучка или аватар;
- сколько стоила видеогенерация;
- сколько стоила пересборка после ошибки;
- сколько материалов реально дошло до публикации;
- какой из них дал заявку, разговор или доверие.

Пока это не посчитано, разговор о «масштабировании контента» остаётся красивой фантазией.

Почему я внедрил Hyperframe

Самая здравая часть пайплайна для меня оказалась не там, где «нейросеть делает магию», а там, где можно вернуть контроль.

Так я пришёл к Hyperframe: HTML как источник видео. Не промпт в чёрный ящик, а управляемая композиция. Текст, блоки, тайминг, анимация, титры, цвет, шрифты — всё можно описать, проверить, пересобрать и воспроизвести.

Для контент-завода это важнее, чем кажется.

Генеративное видео красиво, пока оно одноразовое. Но если нужно делать серию роликов в одном стиле, с понятной типографикой, повторяемыми заставками, субтитрами, схемами и аккуратной подачей, хочется не угадывать каждый раз, а иметь станок.

Hyperframe даёт именно станок:

- один стиль можно использовать снова;
- ролик можно собрать из текста и разметки;
- титры не плывут случайно;
- схемы можно анимировать предсказуемо;
- правки не требуют заново «просить модель сделать красиво»;

- результат можно пересобрать, если изменилась фраза, цвет или логика сцены.

Это не отменяет генеративные модели. Они остаются полезными для идей, сценариев, картинок, черновиков, голосов, отдельных кадров. Но финальная сборка должна быть управляемой. Иначе контент-завод превращается в казино: иногда выпадает красиво, иногда нет, а почему — непонятно.

Где в этом месте полезны аватары

Цифровые аватары выглядят соблазнительно. Можно не сниматься каждый день. Можно быстро выпускать объясняющие видео. Можно масштабировать присутствие эксперта.

Но аватар не решает главную проблему. Он не создаёт мысль.

Если текст пустой, аватар просто произнесёт пустоту человеческим лицом. Если оффер слабый, аватар сделает слабый оффер чуть более гладким. Если нет позиции, получится видео, которое вроде профессионально выглядит, но не цепляет.

Я бы использовал аватары не как замену человеку, а как усилитель регулярности. Например:

- короткие объяснения по уже готовым материалам;
- вводные ролики к чеклистам и PDF;
- ответы на типовые вопросы;
- версии одного объяснения под разные сегменты;
- внутреннее обучение, где важнее ясность, чем актёрская игра.

Для сильных авторских видео человек всё ещё нужен. Не потому что камера священна. А потому что зритель считывает риск, паузу, сомнение, интонацию и личную ставку. Аватар это имитирует, но не всегда несёт.

С голосом у меня получилось так же. Из-за ограниченного доступа к нормальным моделям озвучки приходилось выбирать между плохим синтезом и собственным голосом. Стандартные голоса часто звучат либо пластмассово, либо слишком «дикторски», либо просто не попадают в мою интонацию. Поэтому часть материалов я озвучивал сам.

Я не профессиональный актёр. Особенно это заметно в моём пайплайне Telegram-кружков: круговое видео, поверх него субтитры, фирменный стиль, плотный текст. Заучить страницу А4 и рассказать её дикторским голосом без запинок — отдельная профессия. Я этой профессией не владею.

Но практика — критерий истины. В живом голосе остаются шероховатости: дыхание, паузы, неидеальная подача, место, где мысль догоняет фразу. Для рекламного ролика это может быть недостатком. Для авторского контента — часто наоборот. Это оставляет живой нерв и делает материал честнее.

Поэтому в контент-заводе не всегда нужно стремиться к идеально гладкой озвучке. Иногда лучше честный человеческий голос с субтитрами и ясной мыслью, чем безупречный синтез, от которого пахнет демонстрацией технологии.

Как я теперь смотрю на контент-завод

Я не верю в контент-завод как в машину, которая сама принесёт аудиторию и продажи. Верю в другой вариант: маленький, жёстко ограниченный производственный контур вокруг конкретной задачи.

Например:

1. Раз в неделю собрать 20–50 сигналов: вопросы клиентов, посты конкурентов, новости, комментарии, поисковые запросы.
2. Отобрать не темы, а гипотезы: что болит, у кого, почему сейчас, к какому офферу это ведёт.
3. Выбрать один сильный материал недели.
4. Сделать из него статью, короткий ролик, 3–5 постов, письмо и фрагмент лендинга.
5. Пропустить через редакторский фильтр: звучит ли это как автор или как усреднённый ИИ.
6. Опубликовать не всё, а лучшее.
7. Посмотреть на реакцию: дочитывания, ответы, сохранения, переходы, заявки, качество разговоров.
8. Обновить банк гипотез.

Это медленнее, чем мечта о полностью автоматическом заводе. Зато честнее.

Что можно продавать клиенту

Клиенту не нужно продавать «мы будем генерировать вам контент». Это звучит дёшево и опасно. Генераторов контента уже слишком много.

Продавать можно другое:

- карту трендов и вопросов аудитории;
- банк контентных гипотез, связанных с оффером;
- пайплайн переупаковки экспертного материала;
- серию видео в управляемом стиле;
- прототип цифрового аватара для типовых объяснений;
- систему редакторских правил, чтобы бренд не звучал как нейросеть;
- автоматизацию черновиков, обложек, субтитров и описаний;

- еженедельный отчёт: что сработало, что выкинуть, что проверить дальше.

Это уже не «посты на заказ». Это маркетинговый производственный контур.

Выигрыш этого направления

Для бизнеса выигрыш не в том, что контента стало больше. Его и так слишком много.

Выигрыш в том, что компания быстрее видит сигналы рынка, быстрее проверяет углы подачи, дешевле переупаковывает экспертизу и перестаёт каждый раз начинать с белого листа.

Для эксперта выигрыш другой: можно чаще появляться в поле зрения, не превращая жизнь в бесконечную съёмку. Один сильный смысл может стать статьёй, видео, письмом, чеклистом, фрагментом лендинга и серией коротких постов.

Но финальный фильтр остаётся человеческим.

Если материал нельзя подписать своим именем, его нельзя выпускать. Если ролик выглядит дорого, но не несёт мысли, он всё равно дешёвый. Если автоматизация увеличивает шум, её нужно не масштабировать, а остановить.

Главный вывод этой главы простой:

Контент-завод должен производить не контент. Он должен производить проверяемые гипотезы, которые автор готов подписать своим именем.

Глава 21. Как разработчику продавать ИИ-трансформацию хорошо

Разработчик сегодня может собрать продукт быстрее, чем когда-либо: агент, интеграция, обработка данных, внутренний инструмент, автоматический отчёт, прототип сервиса. Производственная сила выросла резко. Но продажи не появляются сами из факта, что вы умеете делать сложные вещи.

Взрослая продажа ИИ-трансформации начинается не с фразы «я умею внедрять ИИ». Она начинается с точного выбора боли, клиента и первого безопасного шага.

Клиент не покупает модель, агента или API. Он покупает изменение состояния: меньше ручной работы, быстрее ответ, понятнее картина, дешевле операция, меньше ошибок, больше заявок, быстрее проверка гипотез. Поэтому задача разработчика-консультанта — перевести техническую возможность на язык бизнес-результата.

1. Выбрать узкую зону, а не продавать «ИИ вообще»

Плохая позиция:

Я занимаюсь ИИ и автоматизацией.

Слишком широко. Непонятно, кому это нужно прямо сейчас.

Хорошая позиция:

Я помогаю компаниям найти и автоматизировать ручные операции в обработке заявок, отчётах и внутренних данных, чтобы быстрее видеть риски и меньше зависеть от ручного копирования.

Ещё лучше — выбрать конкретный сегмент:

- агентства, которые вручную готовят отчёты клиентам;
- B2B-компании, где заявки теряются между сайтом, CRM и менеджерами;
- онлайн-школы, где материалы и тесты нужно проверять на качество;
- сервисные бизнесы, где руководитель поздно видит состояние заявок;
- компании с большим числом документов, писем и повторяющихся согласований.

Узость не уменьшает рынок. Она делает вас понятным.

2. Найти дорогую ручную петлю

ИИ хорошо продаётся там, где есть повторяемая ручная петля:

1. человек получает входящий сигнал;
2. ищет данные в нескольких местах;
3. классифицирует ситуацию;
4. пишет типовой ответ или отчёт;

5. переносит результат в другую систему;
6. руководитель всё равно не видит картину вовремя.

Такая петля почти всегда стоит денег. Даже если компания этого не считала.

Примеры:

- менеджеры вручную разбирают заявки;
- маркетолог каждую неделю собирает отчёт из рекламных кабинетов;
- руководитель просит сотрудников «скинуть статус»;
- методист вручную проверяет тесты и материалы;
- разработчики тратят часы на первичный разбор багов;
- отдел продаж пишет однотипные ответы;
- клиентская поддержка повторяет одни и те же объяснения.

Хороший вопрос клиенту:

Какая повторяющаяся ручная работа у вас раздражает всех, но до неё всё время не доходят руки?

Там часто лежит первый проект.

3. Продавать не внедрение, а первый артефакт

Большая цифровая трансформация пугает. Диагностический артефакт — нет.

Первый продукт должен быть коротким, понятным и покупаемым без долгого доверия. Например:

«Карта ИИ-возможностей за 7 дней»

Что входит:

- выбрать один бизнес-контур;
- описать текущий процесс;
- найти ручные операции;
- оценить данные и доступы;
- определить, где ИИ может только помогать, а где уже действовать;

- выбрать 2–3 сценария автоматизации;
- оценить риски и цену ошибки;
- предложить первый прототип или следующую итерацию.

Что получает клиент:

- карту процесса;
- список узких мест;
- таблицу возможностей;
- приоритет первого внедрения;
- риски и ограничения;
- план следующего этапа.

Это легко купить, потому что клиент не обязан сразу верить в большой проект. Он покупает ясность.

Маленький кейс: как техническая услуга становится покупаемой

Слабая упаковка: «Сделаю вам ИИ-бота для заявок». Клиент слышит игрушку и сразу думает о рисках: кто будет отвечать, что если бот ошибётся, как это подключать, сколько времени уйдёт.

Сильная упаковка: «За неделю разберу путь входящей заявки: сайт, почта, мессенджеры, CRM, скорость ответа. Покажу, где заявки зависают, где можно делать автоматическую классификацию, где нужен человек, и соберу прототип одного безопасного шага».

В первом варианте продаётся технология. Во втором — снятие неопределённости. Руководителю легче купить второе.

4. Упаковать оффер через боль, цену и безопасный шаг

Формула оффера:

Если у вас [тип процесса] сейчас делается вручную, вы теряете [время/деньги/качество/контроль]. Я за [срок] разберу этот контур, покажу [узкие места и сценарии автоматизации] и передам [конкретный артефакт], по которому можно принять решение: внедрять, отложить или сузить задачу.

Пример:

Если заявки с сайта, мессенджеров и почты обрабатываются вручную, часть лидов теряется или получает ответ слишком поздно. За 7 рабочих дней я разберу путь входящей заявки, покажу, где ИИ и интеграции могут ускорить обработку без риска для продаж, и передам карту внедрения с первым прототипным сценарием.

Это не обещает чудо. И хорошо. Руководители с деньгами не всегда верят в чудо. Они верят в ясный следующий шаг.

5. Показать дивиденды в деньгах и управляемости

Не обязательно сразу считать точный ROI. На первом разговоре часто достаточно показать, где возникает выигрыш.

Четыре простых типа дивидендов:

- **Время:** сколько часов в неделю уходит на ручную операцию.
- **Скорость:** насколько быстрее клиент или руководитель получает ответ.
- **Качество:** сколько ошибок, забытых шагов и разночтений можно убрать.
- **Видимость:** насколько раньше руководитель увидит проблему.

Пример расчёта:

3 менеджера тратят по 5 часов в неделю на ручную подготовку отчётов. Это 60 часов в месяц. Даже если автоматизация снимет половину, компания получает 30 часов высвобожденного времени плюс более быстрые управленческие решения.

Не надо раздувать цифры. Лучше честная грубая оценка, чем красивая фантазия.

6. Делать первый контакт через гипотезу, а не просьбу

Разработчики часто пишут так:

Здравствуйте. Я занимаюсь ИИ и автоматизацией. Могу быть полезен.

Это слабое сообщение. В нём нет наблюдения.

Лучше так:

Я посмотрел, как у вас устроен входящий поток: сайт, форма заявки, мессенджер, несколько направлений услуг. В таких схемах часто теряются лиды и нет быстрой классификации заявок. Я занимаюсь разбором таких контуров: нахожу ручные петли, показываю, где можно подключить ИИ/интеграции, и собираю карту первого внедрения. Если актуально, могу предложить короткий диагностический этап без попытки сразу продавать большой проект.

Это всё ещё холодный контакт. Но он звучит как работающий человек, а не как рассылка.

7. Вести разговор по диагностическим вопросам

На первом созвоне не нужно доказывать, какой вы умный. Нужно понять контур.

Вопросы:

- Где сейчас больше всего ручного копирования или сверки?
- Какие отчёты собираются регулярно?
- Где клиент ждёт ответа дольше, чем должен?
- Какие данные лежат в разных системах?
- Какие решения руководитель принимает вслепую?
- Что повторяется каждую неделю?
- Где ошибка сотрудника стоит денег или репутации?
- Что вы уже пытались автоматизировать и почему не взлетело?
- Какие действия ИИ мог бы делать только после подтверждения?
- Что точно нельзя автоматизировать без человека?

После таких вопросов продажа становится не убеждением, а совместной диагностикой.

8. Продавать этапами

Хорошая лестница продажи:

1. **Диагностика.** Карта процесса, узкие места, возможности, риски.
2. **Прототип.** Один сценарий работает на ограниченных данных.

3. **Пилот.** Сценарий используется в реальном процессе, но под контролем человека.
4. **Внедрение.** Интеграция в рабочую среду, права, логи, регламент.
5. **Поддержка и развитие.** Улучшение, новые сценарии, мониторинг качества.

Так клиенту проще покупать. Он не прыгает в неизвестность. После каждого этапа есть решение: продолжать, остановиться, изменить направление.

9. Делать **proof-of-work**, а не портфолио ради портфолио

Если нет клиентов, не надо ждать идеального кейса. Можно сделать демонстрации на типовых процессах.

Например:

- разобрать публичный сайт и показать, как мог бы работать агент обработки заявок;
- сделать пример автоматического отчёта по рекламным данным на демо-датасете;
- показать workflow проверки учебных материалов и тестов;
- собрать прототип базы знаний для внутренней поддержки;
- записать короткий разбор: «где в этом процессе ручная петля и как её убрать».

Важно: демонстрация должна показывать не “смотрите, какая нейросеть”, а “смотрите, где бизнес теряет время и как это можно превратить в контур”.

10. Закрывать продажу не давлением, а следующим решением

В конце разговора не нужно просить доверить большой проект. Нужно предложить ближайший разумный шаг.

Пример:

По разговору видно три зоны, но начинать со всех сразу не стоит. Я бы взял путь входящей заявки: там низкая цена ошибки, быстрый эффект и понятные данные. Предлагаю первый этап на 7 рабочих дней: карта процесса, сценарии автоматизации, риски и прототип одного шага. После этого будет понятно, есть ли смысл идти во внедрение.

Это звучит спокойно. И именно поэтому продаёт лучше.

Практический шаблон мини-оффера

Для кого: компании, где повторяющиеся операции с заявками, отчётами, документами или внутренними данными делаются вручную.

Проблема: сотрудники тратят время на копирование, сверку, классификацию и типовые ответы; руководитель поздно видит картину; часть ошибок не фиксируется.

Первый этап: диагностика одного процесса и карта ИИ-внедрения.

Срок: 5–10 рабочих дней.

Результат: карта текущего процесса, список ручных петель, 2–3 сценария автоматизации, оценка данных и рисков, рекомендация первого прототипа.

Следующий шаг: пилот одного сценария под контролем человека.

Где здесь дивиденды

Для клиента дивиденд — увидеть, где ИИ может дать выигрыш раньше конкурентов, и начать с безопасного участка.

Для разработчика дивиденд — выйти из роли “рук, которые ждут ТЗ” в роль человека, который находит дорогие ручные петли и собирает цифровые контуры. Это выше по ценности. За это платят охотнее, потому что клиент покупает не абстрактную компетенцию, а снижение неопределённости.

Итог простой:

Взрослая продажа ИИ-трансформации — это не умение уговорить. Это умение найти дорогую ручную петлю, посчитать дивиденд, предложить безопасный первый этап и передать понятный артефакт.

Техническая сила становится видимой только тогда, когда у неё появляется коммерческая рамка.

Глава 22. Как понять, что консультант понимает цифровую трансформацию

Предыдущая глава была для тех, кто продаёт трансформацию. Эта — для тех, кто её выбирает.

Хороший консультант не начинает с обещания «мы внедрим ИИ». Это слишком расплывчато. Он сначала вытаскивает на стол реальность: процессы, данные, людей, ограничения, текущую цену ошибки.

Плохой консультант показывает инструменты. Хороший показывает контур.

Плохой говорит: «Можно автоматизировать всё». Хороший спрашивает: «Где автоматизация безопасна, где нужен человек, где нужен журнал, где нужен лимит?»

Плохой продаёт ощущение будущего. Хороший продаёт артефакт: карту процессов, прототип, регламент, интеграцию, отчёт, рабочий агентный сценарий, проверенный workflow, список рисков и следующий шаг.

В цифровой трансформации важно не количество красивых идей, а способность довести одну выбранную зону до состояния, где она работает без ручного героизма. Поэтому первый этап почти всегда должен быть диагностическим. Не абстрактный аудит «вашей цифровой зрелости», а конкретная карта:

- какие процессы повторяются;
- где есть данные;
- где решения принимаются вручную;
- где сотрудники копируют информацию между системами;
- где клиент ждёт;
- где руководитель не видит картину;
- где можно дать ИИ право только читать;
- где можно готовить черновики;
- где со временем можно дать ограниченное действие.

После такой карты становится ясно, что делать первым. Иногда это агент для обработки входящих заявок. Иногда база знаний. Иногда автоматизация отчётов. Иногда нормальный процесс постановки задач. Иногда вообще не ИИ, а банальная интеграция между CRM и сайтом.

Это важный момент. Настоящая цифровая трансформация не обязана выглядеть футуристично. Она обязана уменьшать трение, ошибки и слепые зоны.

Что должно остаться после работы

После хорошего этапа внедрения у клиента остаётся не только впечатление, что «было интересно». Должны остаться артефакты:

- схема текущего и целевого процесса;
- список узких мест;
- критерии успеха;
- прототип или рабочий сценарий;
- правила доступа и остановки;
- чеклист проверки качества;
- регламент использования;
- список следующих итераций.

Если после проекта ничего нельзя передать, показать, проверить и использовать повторно, это была не трансформация. Это была консультационная дымовая машина.

Матрица первого внедрения

Повторяемость × цена ошибки

низкая цена ошибки

высокая цена ошибки

высокая повторяемость

Внедрять первым

черновики, классификация,
сводки, проверяемые отчёты



высокая повторяемость

Human-in-the-loop

юридическое, финансы,
обещания клиентам

низкая повторяемость

Помощник, не проект

разовые тексты, идеи,
нестандартные задачи

низкая повторяемость

Не автоматизировать

кризисы, стратегия,
необратимые решения

Матрица первого внедрения

Часть V. Практические инструменты

Эти разделы можно читать как рабочие листы перед диагностикой, пилотом или разговором с консультантом.

Как я сам пользуюсь этими инструментами

Эти инструменты не лежат отдельно от моей работы. Я использую их как фильтр перед тем, как предлагать решение.

Сначала я ищу не «где бы применить ИИ», а где в бизнесе есть ручная петля: заявка, отчёт, проверка, подбор, публикация, поддержка, подготовка документа. Потом смотрю на повторяемость и цену ошибки. Если ошибка дешёвая, можно быстрее идти к прототипу. Если ошибка дорогая, нужен человек в контуре, логи, подтверждения и откат.

Так же я смотрю на собственные проекты. В Tech Path Finder важен не сам факт генерации вопросов, а управляемость базы знаний, проверки качества, структура направлений и повторяемый процесс улучшения материалов. В контентных проектах важна не пачка постов, а связка: сигнал рынка → гипотеза → материал → публикация → метрики → следующая итерация. При подготовке этой книги логика была той же: тезисы, структура, редактура, схемы, PDF, проверка.

Поэтому инструменты ниже — не украшение в конце книги. Это моя рабочая линейка. Если задача не проходит через неё, я не спешу называть её ИИ-трансформацией.

Инструмент 1. Матрица первого внедрения

Выбирать первый ИИ-процесс надо не по эффективности, а по сочетанию двух факторов: повторяемость и цена ошибки.

Низкая цена ошибки + высокая повторяемость

Это лучшие кандидаты для старта.

Примеры:

- черновики отчётов;
- классификация входящих заявок;
- подготовка сводок;
- первичный разбор документов;
- генерация вариантов текстов;
- поиск дублей и несоответствий.

Здесь можно быстро получать дивиденды и безопасно наращивать автономность.

Высокая цена ошибки + высокая повторяемость

Автоматизировать можно, но только с человеком в контуре.

Примеры:

- юридические документы;
- финансовые действия;
- обещания клиентам;
- медицинские, кадровые, репутационные решения;
- изменение цен и условий.

Правильный режим: ИИ готовит, человек подтверждает, все действия логируются.

Низкая цена ошибки + низкая повторяемость

ИИ полезен как помощник, но строить сложный контур рано.

Примеры:

- разовая презентация;
- единичный анализ;
- поиск идей;
- подготовка нестандартного письма.

Используйте чат или агента, но не превращайте это в проект внедрения.

Высокая цена ошибки + низкая повторяемость

Это плохая зона для автоматизации.

Примеры:

- стратегическая сделка;
- конфликт с ключевым клиентом;
- критическое юридическое решение;
- разовый кризис.

ИИ может помочь собрать факты, варианты и риски. Решение остаётся у человека.

Простое правило

Начинайте там, где процесс часто повторяется, данные доступны, цена ошибки ограничена, а результат легко проверить. Не начинайте там, где много политики, тумана, устных договорённостей и необратимых последствий.

Инструмент 2. Что не надо автоматизировать первым

Зрелость консультанта видна не только по тому, что он предлагает автоматизировать. Иногда она видна по тому, что он отказывается трогать.

Не надо автоматизировать первым:

- процесс без владельца;
- процесс, который никто не может описать;
- хаос, замаскированный под «у нас так принято»;
- решения с высокой ценой ошибки и без права отката;
- действия с деньгами, договорами и клиентскими обещаниями без подтверждения;
- зоны, где нет доступа к данным;
- процессы, которые ещё не доказали бизнес-ценность;
- то, что проще убрать, чем автоматизировать;
- работу, где проблема не в рутине, а в плохой стратегии.

ИИ не лечит отсутствие процесса. Он ускоряет то, что есть. Если есть ясный контур, он даст рычаг. Если есть бардак, он ускорит бардак и сделает его менее заметным до первой аварии.

Инструмент 3. Форматы работы

Если говорить о консультации и внедрении, разумнее продавать не «цифровую трансформацию вообще», а короткие этапы с понятными артефактами.

1. Диагностика цифрового контура

Срок: 5–10 рабочих дней.

Результат:

- карта текущего процесса;
- ручные петли;
- данные и системы;
- риски;
- 2–3 сценария автоматизации;
- рекомендация первого пилота.

Когда подходит: клиент понимает, что ручной труд мешает, но не знает, с чего начать.

2. Прототип агентного сценария

Срок: 1–2 недели.

Результат:

- рабочий прототип на ограниченных данных;
- сценарий действий агента;
- ограничения;
- критерии проверки;
- решение, стоит ли идти в пилот.

Когда подходит: гипотеза понятна, но нужно быстро проверить техническую и бизнес-реальность.

3. Пилот в реальном процессе

Срок: 2–4 недели.

Результат:

- агентный или автоматизированный сценарий в рабочем контуре;
- human-in-the-loop;
- логирование;
- регламент;
- отчёт по результатам;
- план масштабирования.

Когда подходит: прототип показал пользу, но автономность ещё рано расширять.

4. Сопровождение и развитие

Формат: ежемесячно или итерациями.

Результат:

- поддержка контура;
- улучшение правил;
- новые интеграции;
- мониторинг качества;
- отчёты и backlog следующих улучшений.

Когда подходит: ИИ уже встроен в процесс и должен не деградировать, а становиться лучше.

Инструмент 4. 15 вопросов перед внедрением ИИ

1. Какой конкретный процесс мы улучшаем?
2. Кто владелец этого процесса?
3. Где в нём ручная петля?
4. Сколько времени или денег она съедает?
5. Какие данные нужны для работы?
6. Где эти данные лежат сейчас?
7. Что ИИ должен делать сам?

8. Что ИИ должен только предлагать?
9. Где человек обязан подтвердить действие?
10. Какой результат считаем успешным?
11. Какой результат считаем ошибкой?
12. Как будем проверять качество?
13. Как откатить или остановить действие?
14. Какой артефакт останется после этапа?
15. Какое решение мы примем после первого этапа?

Если на эти вопросы нет ответов, внедрение начинать рано. Сначала нужна диагностика.

Инструмент 5. Карта зрелости: как выбирать и что делать дальше

Если вы выбираете консультанта, подрядчика или внутреннюю команду для цифровой трансформации, не начинайте с вопроса «какие ИИ-инструменты вы знаете?». Это слабый фильтр. Инструменты выучить легко. Сложнее увидеть процесс, риск, метрику и границу ответственности.

Сильные вопросы звучат иначе:

- какой бизнес-процесс мы улучшаем первым;
- где сейчас теряется время, деньги или качество;
- какой артефакт будет передан в конце этапа;
- что ИИ будет делать сам, а что только предлагать;
- какие данные нужны;
- как проверим результат;
- где человек обязан остановить систему;
- что станет повторяемым регламентом, а не разовой демонстрацией.

Цифровая трансформация начинается не с установки инструмента. Она начинается с выбора контура, где цена ручного хаоса уже выше цены внедрения.

Если вы сейчас на уровне чата

Не пытайтесь сразу строить автономных агентов. Начните с дисциплины постановки задачи: роль, контекст, формат, критерии качества.

Если вы уже используете ролевые промпты

Добавьте план перед результатом. Для сложных задач — план для плана.

Если у вас есть агент

Заставьте его работать по plan.md или эквиваленту: цель, шаги, файлы, критерии, проверка.

Если подключаете инструменты

Не подключайте всё. Выберите один повторяемый процесс и дайте агенту ровно нужные доступы.

Если строите workflow

Опишите регламент. Триггер, вход, выход, шаги, ограничения, остановки, лог.

Если делаете образовательные материалы

Не принимайте материал без проверки. Урок должен иметь цель, пример, задание, тест и связь между ними. Тест должен проверять понимание, а не угадывание.

Если делаете бизнес-контур

Начните с чтения и рекомендаций. Потом черновики. Потом действие после подтверждения. Автономность — только после журнала и метрик.

Если хочется много агентов

Сначала спросите: зачем? Если нет разных ролей и независимых перспектив, не множьте сущности. Один агент с хорошим харнесом часто сильнее пяти болтунов.

Послесловие. С чего на самом деле начинается трансформация

ИИ не заменяет стратегию. Он ускоряет то, что уже есть.

Если в компании есть ясный процесс, он даст рычаг. Если есть данные, он поможет быстрее видеть картину. Если есть регламенты, он может выполнять часть работы. Если есть критерии качества, он может проходить проверку.

Но если в компании хаос, ИИ тоже его усилит. Только хаос станет выглядеть современнее: с агентами, дашбордами, автоматическими отчётами и красивыми словами про инновации.

Поэтому цифровая трансформация начинается не с модели и не с модного инструмента. Она начинается с честного взгляда на то, как бизнес реально работает. Где люди переносят данные руками. Где клиент ждёт. Где руководитель слеп. Где знания пропадают. Где ошибки повторяются. Где процесс держится на героизме одного сотрудника.

Только после этого появляется настоящий вопрос:

Какой контур мы готовы улучшить первым — и какой дивиденд хотим получить?

Если вы хотите применить это к своему бизнесу

Эта книга не требует верить в ИИ. Она предлагает более трезвый вопрос: какой процесс уже стоит вам времени, денег, контроля или скорости — и можно ли собрать вокруг него первый управляемый цифровой контур?

Если нужен безопасный вход, начинать стоит не с большого проекта. Начинать стоит с диагностики:

- выбрать один процесс;
- найти ручные петли;
- понять данные и ограничения;

- оценить цену ошибки;
- выбрать 2–3 сценария автоматизации;
- собрать карту первого пилота.

После такой диагностики решение становится спокойнее. Иногда ответ будет: «внедрять». Иногда: «рано». Иногда: «сначала убрать хаос в процессе». Все три ответа полезны, если они экономят месяцы имитации.

Рабочие форматы простые: диагностика цифрового контура, прототип агентного сценария, пилот в реальном процессе, сопровождение и развитие. Главное — не покупать впечатление от демо. Покупать нужно артефакт, который помогает принять следующее решение.

Финальная формула

Цифровая трансформация с ИИ — это путь от ответа к действию, от действия к управляемому процессу, а от процесса — к дивидендам.

Сначала ИИ говорит и помогает думать. Потом действует и ведёт процессы. Потом становится операционным слоем бизнеса. Потом — интерфейсом, оптимизатором, участником рынков. Потом — когнитивным слоем рядом с человеком. Потом инфраструктурой.

Но на каждом уровне остаётся один и тот же закон:

ИИ усиливает систему, в которую его встроили.

Если система честная, он усилит ясность. Если система мутная, он ускорит мутность. Если есть дисциплина, он даст рычаг. Если есть самообман, он даст ему красивую упаковку.

Поэтому зрелое внедрение ИИ начинается не с выбора модели.

Оно начинается с неприятного вопроса:

Что именно мы хотим доверить машине — и как поймём, что она не халтурит?

— Константин Потапов

potapov.me