

ANTHROPIC · 400 000 СЕССИЙ

# Знание дела ПРОТИВ **кода**

Почему с AI-агентом эксперт без кода обгоняет  
вайбкодера. Мысли о профдеформации.



**Ilya Utov**

@gorilla\_under\_hood · Under the Hood

Anthropic выложила исследование на 400 тысячах рабочих сессий с AI-агентом. И показала на данных то, к чему мы с командами шли пару лет на ощупь. Заодно подсветила, почему привычный спор «нужны ли ещё разработчики» с самого начала поставлен криво.

Примерно два года я так или иначе живу с агентским кодингом. Принял не сразу, это отдельная история со своими стадиями. Но в какой-то момент дошло: проблема была не в тупом ИИ, который косит под программиста. Проблема была в нас. В людях с профессиональной деформацией.

Качество ИИшного кода я тут отброшу, товарищам из индустрии привет, вы и сами расскажете, с какими косяками воюете каждый день. Речь про другое.

#### **В ЭТОМ РАЗБОРЕ**

- **Где вайбкодеры правы, а где подменяют тезис**
- **Почему эксперт без кода обгоняет инженера**
- **Что нашла Anthropic на 400 тысячах сессий**
- **Цифры: новичок против эксперта**
- **Что сдвинулось за семь месяцев**
- **Слабые места исследования**
- **Что с этим делать**
- **Так нужен ли инженер**

# Где вайбкодеры правы, а где подменяют тезис

Средний вайбкодер продвигает повестку: разработчики не нужны. Долька правды есть. Только правда не в том, что не нужен разработчик. Правда в том, что сам навык кодирования, именно кодирования, а не инженерии, стал не так важен при работе с агентами.

## КОДИРОВАНИЕ

«Как написать»

Синтаксис, реализация

Это забирает агент

## ИНЖЕНЕРИЯ

«Что и зачем строить»

Какой ценой, где сломается

Это остаётся за человеком

Возьмём владельца бизнеса, продакта, стартапера, любого профессионала, для которого код побочный ресурс, а своё дело он знает вдоль и поперёк. На внутренних задачах он обойдёт любого вайбкодера и любого не продукто-ориентированного программиста. Ему не нужен хайлоад и фронтир. Ему нужно решить задачу, которую он видит как цель.

# Почему эксперт без кода обгоняет инженера с деформацией

---

У меня за плечами 13 лет инженерии. Любая сложная задача в голове мгновенно разворачивается в цепочку: тут сложность, тут нагрузка, тут безопасность, тут вендоры, а тут лицензии. Полезный рефлекс в больших системах. И мешающий, когда задача на самом деле простая.

У человека, который понимает свою тему, этой деформации нет. Зато есть предельная ясность по своей задаче. Условный видеодвижок, который штампует однотипные ролики, не вызывает у него жжения в районе головы по поводу «как». Для него результат это просто видео. Не движок, не пайплайн, не масштабирование.

Для меня это движок с кучей слоёв и абстракций, который должен нарисовать любое видео. Для него, просто видео. Вот в этом и прячется скрытый буст, который теперь подтверждён данными.

# Что именно нашла Anthropic

Около 400 тысяч сессий от 235 тысяч человек за семь месяцев, с октября 2025 по апрель 2026. Данные обезличены, переписки руками никто не читал. В типичной сессии человек ставит цель, агент выбирает путь. Ровно то самое разделение: инженерия человеку, кодирование агенту.

70%

**Человек решает ЧТО**

цель, подход, что считать готовым

/

80%

**Агент решает КАК**

какие файлы, какой код, какие команды

И ядро: пользу усиливает не владение кодом, а понимание задачи. Причём экспертиза привязана к конкретной задаче, а не к должности или диплому.

## ЭКСПЕРТИЗА = ТРИ ВЕЩИ, НЕ ДОЛЖНОСТЬ

- 1 Точность формулировки**, что именно нужно на выходе
- 2 Что просит проверить**, знает слабое место агента
- 3 Кто кого поправляет**, человек агента или агент человека

Бухгалтер без Python, точно задающий правила сверки, для своей задачи эксперт.  
Сильный разработчик в незнакомом языке, новичок.

# Цифры, которые стоит запомнить

Чем глубже человек понимает задачу, тем больше работы агент делает по одной команде.

## ОДНА КОМАНДА: НОВИЧОК ПРОТИВ ЭКСПЕРТА

5 → 12

действий агента (×2)

600 → 3200

слов результата (×5)

Тот же инструмент, та же модель. Разница только в том, кто и как ставит задачу.

## ПОДТВЕРЖДЁННЫЙ УСПЕХ СЕССИИ

15%

### Новичок

доводит дело до твёрдого результата

28-33%

### Средний и выше

понимание задачи примерно удваивает шансы

7 п.п.

### Разрыв профессий

все крупные группы в 7 пунктах от инженеров, менеджеры даже выше

Бросают сложную сессию: новички 19%, остальные 5-7%. Опытный не уходит после первой ошибки.

# Что сдвинулось за семь месяцев

Структура работы заметно поменялась. Отладки стало почти вдвое меньше, а на освободившееся место пришло то, что вокруг кода: запуск программ, разбор данных, подготовка документов.

**СЕМЬ МЕСЯЦЕВ: ОКТЯБРЬ 2025, АПРЕЛЬ 2026**

**33 → 19%**

сессий на отладку (почти вдвое)

**+27%**

ценность средней задачи

Рутину агент забирает, ставки растут у тех, кто умеет ставить задачи посложнее.

# Где я честно притормозлю

Чтобы это не звучало как проповедь, назову слабые места, тем более что Anthropic их сама признаёт.

## ГДЕ Я ЧЕСТНО ПРИТОРМОЗЛЮ

1

### Это данные Anthropic про свой продукт

вывод «и непрограммист справится» им удобен

2

### Успех мерян классификатором

по переписке, не по реальному результату в бизнесе

3

### Горлышко сместилось к дистрибуции

сделать первый результат и продать его, разные навыки

Вывод не «всем будет легко». Инструмент усиливает понимание: у того, кто знает задачу, AI раскрывается, у того, кто не знает, выдаёт мало.

# Что с этим делать

---

Если вы владелец бизнеса и думаете, кого посадить работать с AI, не ищите того, кто умеет кодить. Ищите того, кто глубже всех понимает конкретную задачу. Если вы сами и есть этот человек, вкладывайтесь в три вещи.

## ТРИ ВЕЩИ, В КОТОРЫЕ ВКЛАДЫВАТЬСЯ

- 01 **Формулируйте точно.** Не «сделай отчёт», а «сведи эти данные по таким правилам, поймай дубли, на выходе хочу вот это»
- 02 **Решайте заранее, что проверять.** Эксперт знает слабое место агента и просит показать именно его
- 03 **Не сдавайтесь на первой ошибке.** Почти весь разрыв с новичком, это умение поправить и направить заново

И не надо становиться гуру. Главный скачок происходит на переходе от новичка к среднему уровню, дальше прибавка маленькая. Достаточно нормально разбираться в своём деле.

# Так нужен ли инженер

Честно: нужен. Для коммерческого сервиса на внешнюю аудиторию всплывает куча вещей, где нужны насмотренность и профессионализм: безопасность, нагрузка, поддержка. Здесь без инженера нельзя. А чтобы протестировать идею или закрыть внутреннюю задачу, владелец с пониманием дела дойдёт до первого результата сам.

## ВАЙБ-КОДИНГ

Тыкать в кнопки

Принимать на веру то, чего не понимаешь

Не отвечать за результат

## АГЕНТСКИЙ КОДИНГ

Глубоко понимать свою задачу

Направлять агента осознанно

Отвечать за результат

Разработка ПО изменится сильно. Мы движемся к AI-инженерам, продуктовым инженерам, forward deployment инженерам. Узкое горлышко смещается с кодирования на понимание задачи и умение собрать вокруг неё систему.

Я не принимаю вайб-кодинг, когда тыкают в кнопки и верят на слово. Я топлю за агентский кодинг, когда человек с пониманием задачи направляет агента и отвечает за результат. В этой разнице и есть вся суть исследования.

# Under the Hood

Разбираю, что реально работает с AI-агентами в бизнесе: метод важнее инструмента. Если этот разбор зашёл, в канале такого больше.



**Подпишись: @gorilla\_under\_hood**



наведи камеру