

ВАЙБКОДЕР

как подружиться с ИИ



Go • Спеc • Практика

Сергей Степовой

**Вайбкодер, или Как
подружиться с ИИ**

«Автор»

2026

Степовой С.

Вайбкодер, или Как подружиться с ИИ / С. Степовой — «Автор»,
2026

Книга не про «секретные промпты». Она про ментальную модель и маршрут: LLM, чат, IDE, агент, SDD; как не сжечь лимиты; где ИИ ускоряет, а где опасен; как ревьюить код от ИИ и работать в команде. Сквозной пример — минисервис "Task keeper" на Go (HTTP API, тесты, spec).

© Степовой С., 2026

© Автор, 2026

Содержание

Манифест «Вайбкодера»	5
Глава 1. Вайбкодинг: что это и чем точно не является	7
Зачем эта глава	7
Определение (рабочее, без магии)	8
Чем вайбкодинг НЕ является	9
Три закона (они будут повторяться — специально)	10
Мини-история: два джуна и один Task keeper	11
Кому подходит вайбкодинг	12
Что вы получите, если дочитаете книгу	13
Домашка (15 минут)	14
Что дальше	15
Глава 2. LLM, чат, copilot, агент: одна схема без боли в голове	16
Конец ознакомительного фрагмента.	17

Сергей Степовой

Вайбкодер, или Как подружиться с ИИ

Манифест «Вайбкодера»

ИИ в 2026 — это не «будущее», это ваш коллега, который уже сидит рядом. Он работает быстро, говорит уверенно и иногда несёт такую дичь с таким лицом, что хочется выдать ему доступ в прод из жалости. Не надо.

Эта книга не про «секретный промпт, который сделает вам стартап». Она про то, как **не потерять мозг**, когда у вас появился инструмент, который умеет писать код быстрее, чем вы успеваете его прочитать.

* * *

Мы здесь не за магией, а за результатом

ИИ — не оракул. Он не «знает», он угадывает. Иногда гениально, а иногда — как стажёр после трёх энергетиков.

ИИ — партнёр по черновикам. Он отлично делает «первую версию», «черновик», «варианты». А вот «истину», «правильно» и «в прод» — это всё ещё ваша зона ответственности.

Вайбкодинг — это дисциплина, а не настроение. «Вайб» — не когда вы просите «сделай красиво», а когда у вас выстроен процесс: контекст → итерации → проверка.

* * *

Три закона вайбкодера

ИИ ускоряет черновик, но не снимает ответственность. Ship'ите вы, не модель.

Качество = ваш контекст + ваша проверка. Не «умность промпта», не «новейшая версия LLM модели» и не «самоуверенность ответа».

Инструменты меняются, цикл остаётся. Задача → черновик → ревью → тест → решение.

* * *

Как выглядит дружба с ИИ (а не токсичные отношения)

Дружба — это когда у вас есть:

Границы. ИИ — да: идеи, черновики, шаблоны, тест-кейсы, рефактор по ТЗ.

ИИ — нет: «держи ключи», «реши за меня домен», «потрогай прод», «крипту напиши сам».

Ритм. Маленькие шаги. Короткие диффы. Частые проверки.

Большие запросы = большие сюрпризы. Обычно неприятные.

Трезвость. Если вы не можете объяснить изменение коллеге — значит, вы не поняли, что делаете. Значит, стоп.

* * *

Обещание книги (без маркетинговой пены)

После книги вы:

перестанете путать **LLM, чат, IDE-помощника и агента** (и перестанете спорить о терминах в комментариях),

перейдёте от «пишу в чатик» к системной работе: **контекст, проверка, безопасность, экономия лимитов**,

освоите **SDD (Spec-Driven Development)**— чтобы ИИ делал ровно то, что надо, а не то, что «понял по вайбу»,

научитесь **не сжигать квоту в первый день** и не покупать «ещё пакет токенов» вместо здравого смысла.

* * *

Чего книга не обещает (и слава богу)

Не обещает «сделай фичу без понимания» и «в прод одним промптом».

Не обещает вечный список «топ-10 инструментов на все времена» (интернет уже завален такими археологическими артефактами).

Не обещает, что ИИ перестанет ошибаться. Она обещает, что вы научитесь **ловить ошибки дёшево**, до того как они станут дорогими.

* * *

Правила безопасности (скучно, зато выживем)

Секреты и РП в чат не несём.Никогда. Даже «на минутку».

Большие изменения без тестов и ревью не вливаем.Никогда. Даже «оно же работает».

Деньги, безопасность, прод — только по процессу.ИИ помогает думать, но не подписывает вам релиз.

* * *

Для кого этот манифест

Для джуна и мидл- (dev/QA), которые хотят не хайп, а **чёткий маршрут**:

от первого диалога с ИИ — к SDD и своим агентам — оставаясь профессионалами, а не пассажирами автопилота.

Глава 1. Вайбкодинг: что это и чем точно не является

Зачем эта глава

Слово «вайбкодинг» звучит как мем. И это частично правда — мемы рождаются там, где много хайпа и мало определений.

Но за мемом стоит реальная практика: **как программист/тестировщик использует ИИ каждый день**, не теряя профессионализм.

Эта глава — про смысл слова **в контексте этой книги**. Не про тренды Twitter и не про «замену джунов».

* * *

Определение (рабочее, без магии)

Вайбкодинг — это осознанная работа с ИИ в разработке и тестировании по циклу:
задача → контекст → черновик → проверка → решение

Где:

ИИ ускоряет черновики (код, тесты, спес, объяснения)

выдержите контракт, границы, ревью и ответственность за результат

«Вайб» здесь — не «попросил красиво и надеюсь».

«Вайб» — **ритм маленьких итераций** и привычка не верить уверенному тону модели.

* * *

Чем вайбкодинг НЕ является

Миф	Реальность
«Можно не понимать код»	Понимать надо больше , потому что читать приходится быстрее
«Один промпт — и в прод»	В прод идёт процесс : тесты, ревью, spec
«Это чит»	Чит — это сдать ответственность. Вайбкодинг — разделить : ИИ черновик, вы ship
«Это лень»	Лень — не писать тесты. Вайбкодинг — писать тесты раньше , чтобы не спорить с чатом
«Это только для джунов»	Мидлам и лидам это экономит время на рутине — если есть дисциплина
«Это замена мышлению»	ИИ усиливает мышление, если вы задаёте контракт и проверку

Если узнали себя в левой колонке — добро пожаловать в клуб. Выход есть: глава 7.

* * *

Три закона (они будут повторяться — специально)

Из манифеста:

ИИ ускоряет черновик, не снимает ответственность.

Качество = ваш контекст + ваша проверка.

Инструменты меняются, цикл остаётся.

Запомните их как «стоп-слова», когда хочется нажать «принять весь diff».

* * *

Мини-история: два джуна и один Task keeper

Джун А (вайб без дисциплины):

«Сделай мне REST API для задач на Go»

Получил: gin, postgres, JWT, CRUD, 600 строк. Тестов нет. В прод не пойдёт, в голове — каша.

ДжунВ (вайбкодер):

«Go, net/http, spec 0001, шаг 1: только internal/storage + unit tests»

Получил: маленький пакет, **go test**зелёный, понимает каждую строку.

Разница не в «умности промпта». Разница в **цикле и границах**.

Подробный walkthrough — глава 4.

* * *

Кому подходит вайбкодинг

Хорошо заходит:

джунам и мидл-, которые хотят ускорить рутину и учиться на объяснениях тестировщикам/SDET — черновики кейсов, чек-листы, скелеты автотестов (глава 10)
тем, кто устал от «вечного гугления синтаксиса»

С осторожностью:

если в компании запрет на внешние LLM — сначала политика (глава 5)
если нет тестов и культуры ревью — ИИ ускорит хаос (глава 13)

Плохо заходит:

как оправдание «я не разбираюсь, но залил»
для задач с высокой ценой ошибки без экспертизы (глава 8)

* * *

Что вы получите, если дочитаете книгу

Не «секретный промпт». А систему:

Блок	Главы
Карта инструментов	2–3
Первый опыт	4
Лимиты и безопасность	5
Контекст и цикл	6–7
Куда ИИ не лезть	8
Ревью и QA	9–10
Агенты и SDD	11–12
Команда	13

* * *

Домашка (15 минут)

Прочитайте “Манифест” и выпишите один пункт, который вам «не нравится» — и один, с которым согласны

Сформулируйте для себя: **что вы готовы отдать ИИ**, а что — нет (3 пункта в каждой колонке)

Откройте главу 4: **соберите** Task keeper сами по шагам **или** разберите готовый taskkeeper (<https://github.com/dontagr/taskkeeper>) и тренируйтесь на доп. задачах (тесты, ревью)

* * *

Что дальше

Глава 2— разложим по полочкам: LLM, чат, copilot, агент — без путаницы в терминах.

Глава 2. LLM, чат, copilot, агент: одна схема без боли в голове

Термины: глоссарий

Углубление: глава 11

Сквозно

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.