Бесплатное руководство

AI строит быстро. Как сделать, чтобы он строил то, что нужно

Как методолог работает с AI-агентом. Метод из 955 сообщений моей переписки с Claude за полгода

18 приёмов

по шести шагам работы: от постановки задачи до закрепления

9 ошибок

моих и модели, и правила, которые из них выросли

5 шаблонов

брифа и проверки, плюс чек-листы в шпаргалке

Даниил Евстратов
методолог образовательных продуктов

danilevstratov.ru

Этот гайд для методистов, продактов, авторов курсов и руководителей обучения в онлайн-школах, EdTech и корпоративных академиях. Для тех, кто уже пробовал AI, но работает с ним как с чатом: спросил, получил, переделал.

Здесь мой метод работы с AI-агентом — программой, которая сама ведёт задачу из многих шагов: читает файлы, пишет тексты и код, запускает команды. Мой агент — Claude, AI-ассистент компании Anthropic.

Что вы получите:

— бриф из 8 элементов, после которого первая версия попадает в цель;
— стоп-точки, где AI останавливается и ждёт вашего решения, и проверку — чтобы не платить за переделки и лишние прогоны, то есть запуски задачи от начала до конца;
— лестницу закрепления: как повторяющаяся ошибка AI превращается в правило, которое AI применяет сам;
— 18 приёмов, 9 ошибок с правилами, 4 кейса, шпаргалку, 5 шаблонов, 3 упражнения и словарь терминов.

Чтение займёт 30–40 минут, практика — ещё 30.

Я Даниил Евстратов, методолог образовательных продуктов. Работаю с онлайн-школами, EdTech и корпоративными академиями, собираю AI-агентов для разработки курсов.

Этот гайд собрал скилл из моей собственной переписки с Claude — как именно, в приложении. Скилл — записанная в файл инструкция для повторяющейся задачи: Claude выполняет её каждый раз одинаково.

Содержание

1. Откуда метод 3
2. Модель работы 4
3. Семь принципов 7
4. Приёмы по шагам цикла 10
5. Ошибки и правила, которые из них выросли 21
6. Кейсы 25
7. Инструменты 29
8. Шпаргалка 30
9. Практика 33
Приложение. Как собран этот гайд 34
Словарь 35
Что почитать 39
Если захотите обсудить 40

1. Откуда метод

Метод собран из моей переписки с Claude и на ней же проверен. С апреля по октябрь 2026 года я написал 955 сообщений в 55 сессиях Claude Code и Cowork. Это приложения, где Claude работает как агент: Claude Code — с кодом и командами, Cowork — с файлами и документами. Сессия — один непрерывный рабочий разговор с Claude. Области — методология, продукты, игры, тексты, автоматизация. Каждый вывод в гайде опирается на цитату из переписки.

Полгода в цифрах

955 моих сообщений	55 сессий	14% сообщений со скриншотами (131 из 955)	84% рекомендаций Claude я принял, когда он предлагал варианты на выбор (63 из 75)
9 правил в памяти	17 скиллов	5 задач по расписанию	

Один человек, один AI-ассистент, шесть месяцев: апрель — октябрь 2026.

Задача по расписанию — задание, которое Claude Code сам запускает в заданное время, без меня.

Сообщений в месяц и что закрепилось

Апрель	121	ничего: ошибки повторялись, но правилами не стали
Май	0	сессий нет
Июнь	30	пока ничего
Июль	101	шпаргалка для переноса проектов на другой аккаунт, CLAUDE.md как инструкция проекта, первые скиллы и задачи по расписанию
Август	180	скилл стиля для моих текстов, рисёрч начал превращаться в скиллы
Сентябрь	210	скиллы для разбора своей практики, первые правила в памяти
Октябрь (до 16-го)	313	7 правил в памяти за неделю, скилл ai-playbook, который собрал этот гайд

Закрепление началось в июле. За 16 дней октября — 313 сообщений, за весь апрель — 121.

Что поменялось за полгода

1. От «сделай» к «сначала оцени». В апреле я собирал ТЗ в другом чате и приносил его Claude с просьбой раскритиковать. К октябрю оценка встала в начало задачи. Стоп-точки (приём 4) и решения от меня до правок (приём 9) стали нормой.

2. От ширины к экономии. В августе и сентябре агенты и параллельные прогоны шли по умолчанию. Агенты здесь — помощники, которых Claude сам запускает на части работы, подробнее в главе 2. В октябре по умолчанию быстрый режим, глубокий — по моему слову, расход прикидываем до запуска. Режим — какая модель работает и насколько глубоко она рассуждает. Поворот случился после трёх дорогих эпизодов в рабочих проектах: агент за пару недель потратил около 2000 долларов, правка одного экрана заняла полтора часа, «дешёвый» прогон съел около 10 млн токенов. Токены — кусочки текста, которые модель читает и пишет. По ним считают лимиты — объём работы, который даёт подписка, — и оплату.

3. От памяти в голове к памяти в файлах. Сначала договорённости жили в чате и терялись при сжатии контекста. Контекст — всё, что Claude держит в голове в текущей сессии. Когда он переполняется, Claude сжимает его в краткий пересказ, и детали выпадают. Потом появились файлы: CLAUDE.md — инструкция проекта, которую Claude сам читает при старте; docs/next.md и STATUS.md — состояние дел для следующей сессии. И 9 правил в памяти — заметках, которые Claude сам подгружает в каждую новую сессию. У каждого правила есть «почему» и дата.

4. От текста к голосу. Медиана моего сообщения — 145 знаков. Но в октябре длинные постановки я наговариваю голосом, монологами на 1–5 тыс. знаков. Метод пережил смену канала, потому что сохранился порядок: проблема → идея → результат → формат.

Что не поменялось

— Цель через «чтобы…» и приглашение спорить — с апреля.

— Проверка руками и скриншоты в сообщениях — с первого дня.

— Критерий готовности и язык ответа весь период редко попадали в бриф. Как это закрыть — в главах 5 и 8.

---

2. Модель работы

Обычно работа с AI-чатом состоит из трёх шагов.

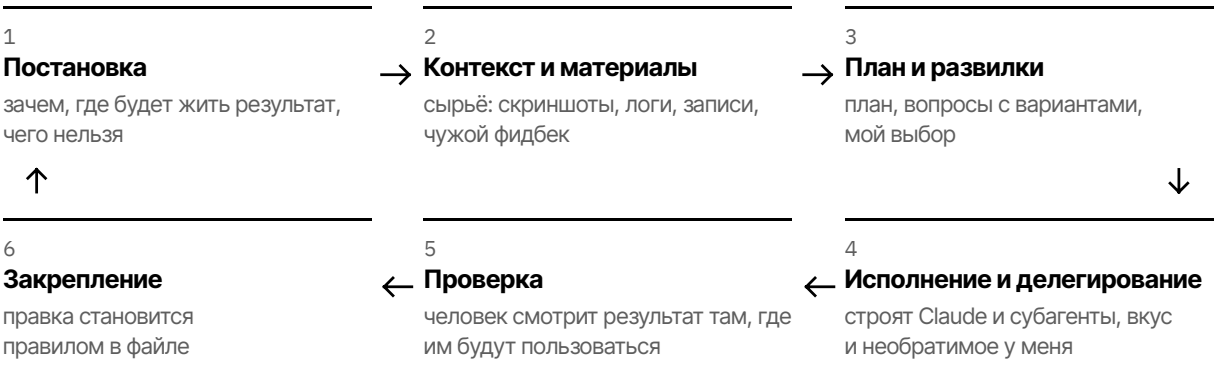
1 Спросить.

2 Получить.

3 Переделать.

У меня шагов шесть. Пропущенный шаг потом всё равно возвращается, чаще всего переделкой.

Цикл из шести шагов



→ снова 1. Постановка, уже с правилами

Ценность копится на шестом шаге: следующий круг начинается с записанных правил.

1. Постановка. Я говорю, зачем нужен результат, где он будет жить и чего нельзя. Цель через «чтобы…» есть примерно в 30 из 50 стартов, которые разобрал скилл. Старт — первое сообщение сессии. Чаще всего ломается критерий готовности: он был в брифе примерно в 6 из 50 стартов. Приёмы 1–4 в главе 4.

2. Контекст и материалы. Я приношу сырьё: скриншоты, логи, записи созвонов, чужой фидбек как есть. Логи — записи о том, что делала программа и где она сбилась. Ломается адрес: около 12 стартов звучали как «мой продукт, который мы сделали», без названия и пути — адреса папки проекта на компьютере (ошибка 8). Приёмы 5–7.

3. План и развилки. Claude предлагает план и вопросы с вариантами. Развилка — момент, когда он показывает варианты и ждёт моего выбора. За полгода было 141 вопрос в 69 развилках. Ломается хвост длинного отчёта: вопрос в конце тонет без ответа. Приёмы 8–10.

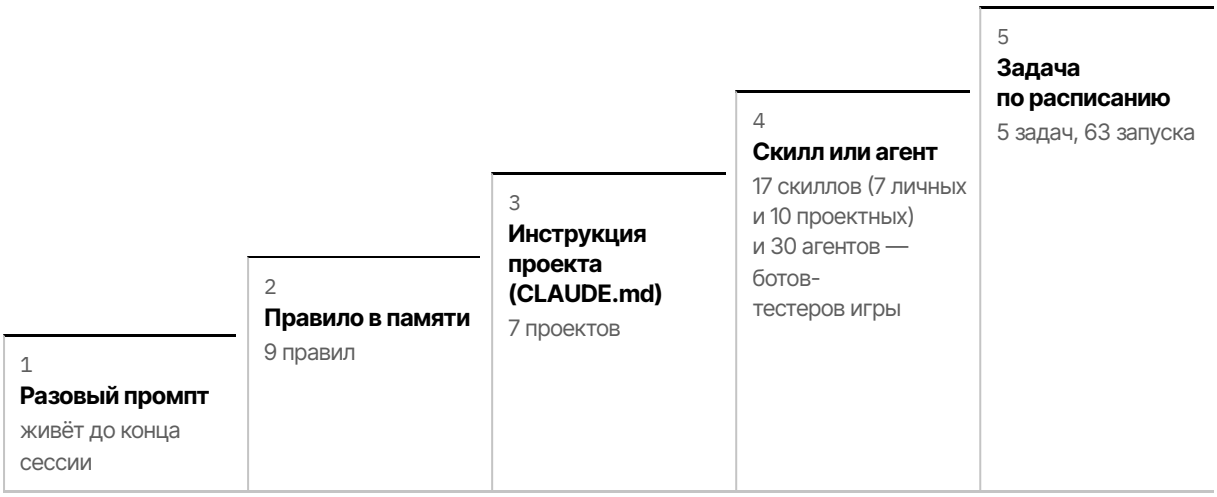
4. Исполнение и делегирование. Claude строит. На рисёрч и ревью он запускает агентов-помощников. В документации их называют субагентами, и они могут работать параллельно. Вкус и необратимые действия остаются у меня. Ломается масштаб. Лишние агенты раздувают расход (ошибка 2). Правка в одном месте ломает рабочее в другом (ошибка 3). Приёмы 11–13.

5. Проверка. Я сам кликаю, играю, открываю на телефоне и пишу, что вижу. Ломается слово «готово»: не меньше 12 раз в пяти проектах сценарий после него не работал (ошибка 1). Сценарий — путь пользователя: что он нажимает и что должен получить. Автотесты — небольшие программы, которые сами проверяют, что продукт выдаёт правильные данные. Клик по кнопке и вёрстку на узком экране они не видят. Приёмы 14–16.

6. Закрепление. Удачную правку я превращаю в правило и прошу записать в файл. Ломается охват: после переименования продукта старое имя осталось в 18 строках сайта. Приёмы 17 и 18.

Лестница закрепления

Правило с шестого шага может подняться выше. Чем выше ступень, тем меньше приходится помнить самому.



Повторяющийся приём превращается в систему, которая работает без напоминаний.

1. **Разовый промпт** живёт до конца сессии или до первого сжатия контекста.
2. **Правило в памяти** — короткое правило с «почему» и датой. Claude сам подгружает его в каждую новую сессию.
3. **Инструкция проекта** — CLAUDE.md в папке проекта: правила, состояние и проверки одного продукта в одном файле.
4. **Скилл или агент** — процедура из нескольких шагов в файле, запускается одной командой.
5. **Задача по расписанию** — процедура, которая запускается сама в заданное время.

Так поднималось правило «баланс считаем математикой». Баланс — насколько игра трудная: сила врагов, цены, запас ресурсов. Сначала была реплика про игру: «математически можно просчитать». Потом плейтест — проверка игры живыми людьми — разошёлся с ботами-тестерами: бот выигрывал 57%, живой человек — 15%. Боты-тестеры — программы, которые сами проходят игру вместо живых игроков. Реплика стала правилом в памяти. Затем в CLAUDE.md игры появилась строка: утверждённые настройки баланса меняются только после моего «баланс хороший».

Строка про «готово» есть только в CLAUDE.md игры: «Не пиши „ощущается хорошо“. Пиши НАБЛЮДАЕТСЯ или НЕ ПРОВЕРЕНО. Вкус решает Даниил». Ошибка 1 случалась в пяти проектах, и правило против неё — о том, как я работаю вообще. Поэтому его короткой версии место в памяти. Пока я её туда не записал.

Когда подниматься выше:

— ошибка или удачная правка повторилась дважды — запишите правилом;

— правило о том, как вы работаете вообще, — в память; правило одного продукта — в инструкцию проекта;

— действие из нескольких шагов повторяется — в скилл;

— действие повторяется по времени — в задачу по расписанию.

---

3. Семь принципов

Принципы объясняют, почему работают приёмы из главы 4.

Принцип 1

Сначала оцени, потом строй

Тезис	Прежде чем строить, Claude оценивает: возможно ли это, насколько сложно, сколько съест и что не так с моей идеей.
Почему	Сборка больше не узкое место: первые версии у меня собираются за 11–50 минут (глава 6). Самая дорогая ошибка теперь — быстро построить не то. Стоп-точка дешевле переделки.
Пример	Перед новой задачей я спросил: «изучи этот плагин для ChatGPT… Скажи, возможно ли воссоздать для Claude для личного пользования». Сначала вердикт о выполнимости, потом сборка.
С чего начать завтра	Разбейте бриф на этапы: оцени → составь план → задай мне вопросы → делай. Попросите остановиться после вопросов.
Где в гайде	приём 4, приём 9, шаблон 1, шаблон 3.

Принцип 2

Сырьё сильнее пересказа

Тезис	Показывайте AI то, что видите сами: скриншот, лог, запись, чужие слова дословно. Свою оценку добавьте одной строкой.
Почему	В сырье Claude видит то, что я бы не смог описать. Скриншоты есть в 14% моих сообщений. Если Claude долго добывает материал сам, быстрее дать его напрямую: текстом, картинкой, файлом.
Пример	Фидбек тестера я принёс как есть и под каждым пунктом дописал своё решение: «Насчёт дорожек согл, что могут запутать, давай мб вместо красной и зелёных полосок сделаем туман войны…» («согл» — согласен, «мб» — может быть; туман войны — часть карты скрыта, пока игрок до неё не дошёл).

С чего начать завтра	Следующую правку отправьте скриншотом и строкой «что вижу → как должно быть».
Где в гайде	приём 5, приём 7, приём 14.

Принцип 3

Вкус мой, техника — тому, кто в ней сильнее

Тезис	Я решаю, что должно получиться. Как это сделать технически, обычно решает Claude.
Почему	Граница проходит по компетенции. Где вопрос в чужой экспертизе, я соглашаюсь: из 75 рекомендаций Claude на развилках я принял 63, это 84%. Своё пишу, когда в варианте спрятаны деньги или лимиты, риск потерять данные, лишняя сложность или выбор вслепую. Необратимое — вход в аккаунт, «Отправить», публикацию — делаю сам или по моему «да».
Пример	Отдаю технику: «Рисуй сам, я не буду рисовать в Procreate: маленькие штуки я точно плохо нарисую». Забираю решение: «история с прежним судьёй мне кажется слишком глубокой и сложной для нашей игры».
С чего начать завтра	Решите до старта, что в задаче ваше: вкус, финальный выбор, необратимые шаги. Остальное отдайте и попросите варианты с рекомендацией.
Где в гайде	приём 2, приём 10, приём 12, глава 5.

Принцип 4

Мои идеи — гипотезы, со мной можно спорить

Тезис	Свою идею я подаю как гипотезу и заранее разрешаю возразить.
Почему	Право возразить окупается. Claude спорит по существу, отклоняет лишнюю работу и признаёт, когда ошибся сам. Сомневаться стоит и в методике проверки (приём 15).
Пример	Про механику игры: «Как будто то, что копейщик теряет в эффективности при склеивании, неправильно. Или ошибаюсь?» Ответ Claude: «Ты прав… я ошибся». С 3 октября в памяти есть правило: работать как напарник и спорить с моими идеями.
С чего начать завтра	Допишите к просьбе: «Это гипотеза, проверь её. Если я ошибаюсь — скажи прямо».
Где в гайде	приём 8, приём 15, шаблон 2.

Принцип 5

«Готово» — когда человек проверил результат там, где им будут пользоваться

Тезис

Зелёные тесты и «выложено» — промежуточный статус. Зелёные тесты значат, что все автотесты прошли без ошибок. «Выложено» — продукт опубликован в интернете, им уже можно пользоваться. Готово, когда человек открыл результат в настоящей среде: в игре, на телефоне, у пользователя.

Почему

Жест, системное уведомление и ощущение от игры Claude сам не проверит. Понятность проверяют только люди (ошибка 6).

Пример

Спрайт — игровую картинку — перерисовали во внешнем генераторе, то есть в сервисе, который рисует картинку по текстовому описанию. Вердикт после проверки в игре: «хорошо в генераторе, мыльно в игре». «Мыльно» значит размыто.

С чего начать завтра

Добавьте в бриф строку «Готово, когда…» с проверкой в настоящей среде: «проверь на узком экране», «кликни каждую новую кнопку». В отчёте попросите назвать, что не проверено.

Где в гайде

приём 14, приём 16, ошибка 1, ошибка 6, шаблон 4, глава 8.

Принцип 6

Быстрый режим по умолчанию, глубокий — по слову

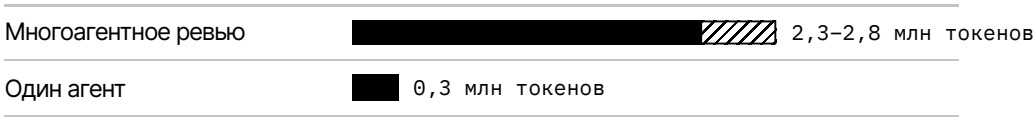
Тезис

Точечную правку Claude делает сам и за один заход. Агентов и глубокое ревью включаю я, словом.

Почему

Ширина не равна качеству. Раньше она шла у меня по умолчанию (ошибка 2). Сравните расход на одно ревью.

Сколько токенов съедает одно ревью



Качество сопоставимое. Платите за глубину, только когда она нужна.

Пример

Из долгой сессии: «А давай, может, в конце проведём тесты? А то долговато получается и очень много токенов тратится: в 15:30 начали, а сейчас почти 6 часов прошло». Чуть позже: «А ты не можешь сам решать, какой режим включать — ультра или поменьше?» Ультра — самый глубокий и дорогой режим. Теперь обычный режим Claude выбирает сам и объявляет строкой перед работой. Глубокий включаю я.

С чего начать завтра

Допишите в бриф: «Работай в быстром режиме. Перед запуском агентов скажи, сколько это примерно съест».

Где в гайде

ошибка 2, глава 7.

Принцип 7

Урок записываю в файл

Тезис	Урок, который остался в чате, пропадёт. Я записываю его в файл правилом — с «почему» и датой.
Почему	Договорённости из чата теряются при сжатии контекста и в новой сессии. Правило из файла Claude прочитает сам и применит без напоминаний.
Пример	Даже отложенную идею прошу записать: «но это пока зафиксируй, потом продумаем, давай пока королеву роя и комаров всех поменяем». За неделю с 30 сентября по 6 октября, в разгар работы над игрой, появились 7 из 9 моих правил в памяти.
С чего начать завтра	Когда AI повторит ту же ошибку второй раз, попросите: «Запиши это правилом: что делать, почему, с какой даты». Перед паузой попросите сводку в файл статуса. Новую сессию начните со строки «Читай `<файл статуса>`».
Где в гайде	приём 13, приём 17, приём 18, глава 2, глава 9.

---

4. Приёмы по шагам цикла

В главе 18 приёмов по шести шагам цикла из главы 2. Каждый встречался у меня минимум в трёх сессиях. Находки поменьше — во врезке «Лайфхаки» в конце главы.

В карточке шесть строк: ситуация, как делаю (с цитатой из переписки), почему работает, где ломается, ступень и «Попробуйте» — одно действие на вашу ближайшую задачу. **Ступень** — докуда приём дошёл по лестнице закрепления:

- **разово** — сработало однажды;
- **привычка** — делаю сам и держу в голове;
- **правило в памяти** — записано правилом в память (ступень 2);
- **CLAUDE.md** — записано в инструкцию проекта (ступень 3);
- **скилл или агент** — упаковано в скилл или в сохранённого агента (ступень 4);
- **расписание** — стало задачей по расписанию (ступень 5).

В лестнице из главы 2 «привычки» нет: это разовый промпт, который я каждый раз повторяю сам.

Больше половины приёмов главы стоят на ступени «привычка»: работают, пока я о них помню. Как поднять приём выше — в главе 2.

Постановка

1 Задаю одно жёсткое ограничение

Ситуация	Нужен небольшой инструмент, и рука тянется писать ТЗ на страницу.

Как делаю	Пишу одну фразу о том, чего продукт делать не должен, и пример поведения. Бриф бота, который присылает погоду из места, где хуже: «Хочу сделать бота в тг. Чтобы он не ел денег, только на сервере лежал (в плане токенов нейросетей и т.д.)». «Тг» — Telegram, бот — программа в мессенджере, которая сама принимает и присылает сообщения. Сервер — компьютер в интернете, который работает круглосуточно. Примерно через 12 минут бот работал: без нейросетей, на бесплатном сервисе погоды.
Почему работает	Ограничение отсекает варианты раньше, чем Claude начнёт строить. «Не ел денег» убрало нейросети и платные сервисы, и архитектура сложилась сама.
Где ломается	Ограничение, сказанное один раз, теряется, когда продукт растёт (ошибка 3). Помогает инвариант — свойство продукта, которое проверяют после каждой доработки.
Ступень	Привычка. Инварианты я пока не превратил в пункты проверки.
Попробуйте	Следующий небольшой бриф начните с одной фразы-запрета и одного примера.

2 Называю антипример и антицель

Ситуация	Результат про вкус: ролик, лендинг, текст. «Сделай красиво» даёт среднее по больнице.
Как делаю	Называю конкретную плохую версию, остальное оставляю Claude. Про промо-ролик своего приложения: «Насчёт конкретного плана реши сам… Единственное, хочу, чтобы не слопно выглядело — не как типичный стоковый ИИ-ролик в стиле Apple, но чтобы продуктово и стильно, и про кор-фичи». Слоп — шаблонный AI-контент, который узнают с первого взгляда. Кор-фичи — главные функции продукта.
Почему работает	Антипример короче описания идеала и сразу задаёт планку. «Стильно» каждый понимает по-своему. Стоковый AI-ролик в стиле Apple видели все. План при этом Claude выбирает сам.
Где ломается	Антипример без направления оставляет Claude угадывать. Поэтому в той же фразе я пишу, каким результат должен быть: «продуктово и стильно».
Ступень	Привычка. «Чтобы не было ИИ-слопа» — сквозной критерий моих задач.
Попробуйте	Допишите к заданию две строки: «точно не хочу: …» и «хочу, чтобы…».

3 Даю критерий-ощущение вместо числа

Ситуация	Нужно настроить то, что человек чувствует: сложность, темп, баланс.

Как делаю

Описываю, что человек должен почувствовать или понять. Числа подбирает Claude. Про баланс моей игры «Скрижаль»: «Если ты их не убил и провалил защиту — значит, надо было об этом думать. Это объяснимая логика, почему ты проиграл. А сейчас я вроде всё сделал, а всё равно проиграл». После правки: «не слишком легко, не слишком тяжело, можем дальше».

Почему работает

Ощущение — это вкус, числа — техника, и каждый делает свою часть (принцип 3). Методисты так же формулируют образовательный результат: через то, что человек сделает и поймёт.

Где ломается

Критерий легко забыть: он есть примерно в 6 из 50 стартов. И проверить ощущение может только человек. Поэтому в CLAUDE.md игры записано правило «НАБЛЮДАЕТСЯ или НЕ ПРОВЕРЕНО» (глава 2).

Ступень

CLAUDE.md игры.

Попробуйте

Допишите в бриф строку «готово, когда `<кто>` почувствует или поймёт `<что>`».

4 Ставлю стоп-точки: оценка → план → вопросы ко мне → сборка

Ситуация

Задача большая: продукт, агент, прототип.

Как делаю

Нумерую этапы и говорю, где Claude ждёт меня. В проекте для заказчика: «1) посмотри, оцени, провалидируй 2) составь план 3) окни об меня 4) запусти агентов, реализуй». «Провалидируй» — проверь, выдерживает ли идея критику. «Окни об меня» значит «сверься со мной».

Почему работает

Это принцип 1 в действии: стоп-точка стоит одно сообщение, переделка — новую сборку. Скорость почти не страдает: моя игра для вечеринки дошла от брифа с оценкой концепта до деплоя, то есть выкладки в интернет, за 50 минут.

Где ломается

Без границы между оценкой и сборкой вопрос «можно ли» разрастается. Одна моя сессия началась с «расскажи, можно ли это сделать» и дошла до 129 сообщений (ошибка 9). Просите на оценке вердикт: можно ли, насколько сложно, сколько съест.

Ступень

Привычка. Цикл «оцени → план → реализуй» стоит и в скилле, который собрал этот гайд.

Попробуйте

В следующей большой задаче допишите после пункта про план: «дальше не иди, жди моего „да“». Полная версия — шаблон 1.

Где Claude ждёт меня

1		2		3		4
Оценка «посмотри, оцени, провалидируй»	→	**План** шаги и точки, где я проверяю	→	**Стоп** «окни об меня»: вопросы ко мне, жду «да»	→	**Сборка** «запусти агентов, реализуй»

Я вмешиваюсь до сборки, пока правка стоит одно сообщение.

Контекст и материалы

5 Приношу сырьё целиком

Ситуация — Что-то сломалось или пришёл отзыв, и хочется пересказать своими словами.

Как делаю — Приношу исходник: скриншот, текст ошибки, логи, запись созвона, чужой фидбек как есть. Добавляю строку: что проверял и чего не смотрел. Логи забега — одной партии в моей игре от начала до конца — я прислал с подписью: «А я хотел проверить карту, что враги падают в яму, но вроде никто не упал». По таким подписям Claude отделяет мои эксперименты от замеров сложности.

Почему работает — Пересказ — уже интерпретация, и в нём теряется деталь, которая часто и оказывается причиной.

Где ломается — Сырьё в неудобном формате съедает время. Однажды Claude потратил 169 шагов в браузере (кликов, прокруток, вводов текста) и 44 чтения файлов на библиотеку материалов. Три PDF при этом пришли пустыми. Сдвинуло работу одно сообщение: «Давай ещё раз, экспортнул пнгшками. Там сразу всё». То есть я сохранил материалы картинками PNG. Если модель долго добывает материал, дайте его напрямую.

Ступень — Привычка.

Попробуйте — Следующую ошибку или чужой отзыв пришлите как есть, со строкой «проверял…, не смотрел…».

6 Разбираю образец по трём корзинам

Ситуация — Нашёлся референс, то есть образец, который нравится: лендинг, курс, ролик.

Как делаю — Прошу разложить его на три корзины: «Мне оч понравился лендинг — и простота дизайна, и наличие бренд-персонажа, и вординг, и структура… Давай изучим его и декомпозируем — что классно, что нам подходит, что не подходит». Вординг — формулировки текстов. Методистам это знакомо по разбору конкурентов с графой «что можно взять себе».

Почему работает — Каждый элемент проходит через вопрос «подходит ли нам», и из референса выходит список решений. Пустая корзина — тоже ответ: однажды Claude прямо сказал, что из набора готовых элементов интерфейса в моё приложение для Mac переносить нечего.

Где ломается — Без строки «берём это, не берём то» референс уводит в сторону. Видео про графику, которую генерирует код, я прислал ради простого генератора спрайтов. Claude прочитал его как «скелеты и части», то есть как анимацию персонажа из подвижных частей. Картинку и видео дублируйте словами (ошибка 4).

Ступень	Привычка, правило пока не записано.
Попробуйте	К следующему референсу допишите три строки: что классно, что берём, чего не берём.

7 Калибрую содержание контрпримером из жизни аудитории

Ситуация	Claude выдал примеры, вопросы или карточки, и они мимо аудитории.
Как делаю	Беру один промах и ставлю рядом то, что этим людям ближе. Про игру для вечеринки с друзьями: «Например, по чему ты скучаешь — интернет без рекламы: не актуально… это сильно более актуальный культурный контекст для моих друзей». Claude прогнал через вопрос «резонирует ли» все 25 наборов карточек.
Почему работает	Из одного контрпримера Claude выводит критерий и применяет его ко всему набору. Методист делает то же с кейсами курса: пример не из жизни ученика его не цепляет.
Где ломается	Контрпример хорош настолько, насколько вы знаете аудиторию. Друзей я знаю. Если аудиторию вы знаете хуже, контрпримеры принесут живые люди: их реплики, дословно (принцип 2).
Ступень	Привычка.
Попробуйте	Найдите в ответе AI пример, который ваша аудитория не узнает. Поставьте рядом свой и попросите проверить весь набор.

План и развилки

8 Прошу спорить: мои идеи — гипотезы

Ситуация	У меня есть идея или готовое ТЗ, и я в нём не уверен. Или уверен, что ещё опаснее.
Как делаю	Разрешаю критику заранее, прямо в постановке: «(прочелленджи и провалидируй идею)», то есть проверь на прочность. Свою версию заканчиваю выходом: «Или ошибаюсь?» (принцип 4).
Почему работает	«Сделай» ставит Claude в роль исполнителя, приглашение спорить — в роль напарника. В проекте для заказчика я подал найденное в методической модели противоречие как гипотезу. Claude назвал его ошибкой категории, то есть сравнением несравнимого, и проверка выросла в системную: 101 линия модели, автоматических проверок стало 80 вместо 74.
Где ломается	Спор без базы скатывается в общие слова. Перед ревью концепта игры я попросил Claude сначала собрать практики геймдизайна. Вышла база на 18 тыс. слов (кейс «Скрижаль», глава 6), и спорить стало о чём.
Ступень	Правило в памяти: работать как напарник и спорить с моими идеями.
Попробуйте	Закончите следующую постановку строкой «если я ошибаюсь, скажи прямо» (шаблон 2).

9 До правок прошу выписать решения, которые нужны от меня

Ситуация — В пачке правок есть развилки. Claude может выбрать сам или спросить посреди работы.

Как делаю — В проекте для заказчика я попросил: «а выпиши в чат решения, которые от меня нужны перед фиксами», то есть перед исправлениями. Отвечаю одним сообщением по номерам и тут же докидываю следующую задачу.

Почему работает — Claude не чинит наугад. Я отвечаю на всё за один заход, пока задача свежа в голове, и сразу выношу лишнее за рамки задачи.

Где ломается — Вопрос в хвосте длинного отчёта тонет. Однажды я такой пропустил, и потом игра «удивила» меня: запустилась сразу в обучение. Просите Claude нумеровать вопросы и заканчивать отчёт строкой «сделано / не сделано / ждёт меня».

Ступень — Привычка. Строку статуса в конце отчёта я пока не закрепил.

Попробуйте — Перед следующей пачкой правок напишите: «выпиши решения, которые нужны от меня, с вариантами и твоей рекомендацией» (шаблон 3).

10 Прошу варианты листом и выбираю по номерам

Ситуация — Решение про вкус: шрифт, стиль, спрайт, формулировка. Словами его не описать.

Как делаю — Прошу варианты рядом, выбираю по номерам, склеиваю лучшее. Про пиксельный шрифт: «Ты мне покажешь, нарисуешь примеры пары-тройки текстов в таком же стиле, но с x2, x3, x4 по количеству пикселей». Здесь x2, x3, x4 — пиксели в два, три и четыре раза крупнее. Про защитников игры хватило одной строки на шесть решений: «1) а 2) а 3) в 4) в 5) а 6) в».

Почему работает — Выбирать быстрее, чем описывать. Описать шрифт словами умеют дизайнеры и поэты, узнать нужный из трёх может любой. Вкус остаётся за мной.

Где ломается — Вариант с мокапа, то есть макета, может не выдержать настоящего объёма. Шрифт Tiny5 выбрали на примерах, на реальных экранах он рябил. Финальный выбор делайте там, где результатом будут пользоваться (принцип 5).

Ступень — Привычка.

Попробуйте — В следующем вопросе про вкус попросите три пронумерованных варианта и ответьте одной строкой.

Исполнение и делегирование

11 Прошу промпт для другого инструмента

Ситуация — Нужна картинка, макет экрана или прототип. Профильный сервис сделает их лучше Claude.

Как делаю

Claude пишет промпт для Claude Design, Lovable или генератора картинок. Claude Design — инструмент Anthropic для макетов и прототипов. Lovable — сервис, который собирает сайт или приложение по текстовому описанию. Генерирую я, результат приношу обратно и прошу собрать лучшее из вариантов. Пока я генерирую, Claude занят другим: «можешь сначала просто дать подробный промпт-документ для Lovable… а в это время ты параллельно проверишь наши материалы».

Почему работает

Claude становится штурманом между инструментами. Промпт пишет тот, кто держит контекст проекта. Рисует тот, кто умеет рисовать.

Где ломается

На стыках. Картинка по ссылке может не дойти до Claude, поэтому идею стоит продублировать словами. Имя известной игры в промпте генератор понимает буквально (ошибка 4).

Ступень

Привычка.

Попробуйте

Попросите AI написать промпт для профильного инструмента и предупредите, что принесёте результат обратно. Каркас — шаблон 5 в главе 8.

12 Необратимое делаю сам

Ситуация

Claude умеет заполнять формы в браузере, публиковать сайты и отправлять код на сервер. Часть этих действий не отменить.

Как делаю

Точки невозврата оставляю себе: вход в аккаунты, галочки согласия, кнопку «Отправить», публикацию и push — отправку кода на сервер. Их делаю я, или Claude ждёт моего «да». Однажды после моего входа в аккаунт Claude прошёл 194 шага в браузере и остановился перед галочкой согласия и кнопкой «Отправить».

Почему работает

Доверие у меня высокое: 86% сообщений написаны в режиме полной автономии, Claude не спрашивает разрешения на каждый шаг. Стоп-кран при этом живой: 28 раз я прерывал работу на ходу. Человеку остаются только точки невозврата.

Где ломается

Ожидание «да» копит долг: пока мы обсуждали контент игры, накопилось около 20 неотправленных коммитов — сохранённых версий кода. Выход — локальный коммит после каждой правки, то есть сохранение версии у меня на компьютере, без отправки на сервер. На сервер версии уходят пачкой, по моему слову. Это правило я пока не записал (ошибка 3).

Ступень

Привычка, Claude держит её сам.

Попробуйте

Выпишите три действия, которые в вашей работе нельзя отменить, и добавьте в инструкцию проекта строку «эти шаги — только после моего „да“».

13 Разношу работу по контурам и сессиям

Ситуация

Проект живёт неделями. Чат разрастается, темы смешиваются, и Claude путает старые договорённости с новыми.

Как делаю

Делю работу по контурам и сессиям. Контур — место, где идёт работа: в облаке или на моём компьютере. Облачная сессия идёт на сервере Anthropic, без доступа к моему компьютеру. Она берёт бумагу: тексты и вайрфреймы, то есть схемы экранов. Сессия на маке идёт на моём компьютере, и Claude видит его файлы и программы. Она берёт интерфейс и графику: «потому что в облаке вроде не можем с UI работать» (UI — интерфейс). Каждая итерация — новая сессия с первой фразой «Читай docs/next.md»: в этом файле состояние проекта и следующий шаг. Отдельную тему выношу из чата: «Может, лучше этот чатик форкнуть? А то мы тут LinkedIn'ом занимаемся и всё смешается». Форк — ответвление чата — при этом копирует всю историю, поэтому для новой темы лучше чистая сессия (ошибка 9).

Где живёт какая работа

Облачная сессия	тексты и вайрфреймы
Сессия на маке	интерфейс и графика
Новая итерация	новая сессия, первая фраза «Читай docs/next.md»
Новая тема	новая сессия: форк копирует всю историю

Договорённости живут в файлах, и каждая новая сессия начинает с них.

Почему работает

У каждой сессии чистый контекст. Договорённости лежат в файле, и сжатие контекста их не съест.

Где ломается

Когда я сам нарушаю правило. Самая длинная сессия, по игре, дошла до 159 сообщений и 18 сжатий, хотя Claude несколько раз советовал начать новую. Договорённости сверяли по README — файлу с описанием проекта.

Ступень

Правило в памяти: «каждая итерация в новой сессии, передача через docs/next.md». Пока правило написано только для игры.

Попробуйте

Следующую итерацию начните в новой сессии с фразы «Читай `<файл со статусом>`».

Проверка

14 Пишу «что вижу → как должно быть»

Ситуация

Claude отчитался, вы открываете результат, и что-то не так. Хочется написать «не работает» или сразу угадать причину.

Как делаю

После каждой сборки играю или кликаю сам. Пишу ощущение и цифру-симптом: «В предыдущую версию мне было играть сильно интереснее, потому что там был челлендж… У меня к 20-й волне было 2000 свободного золота, мне нечего было покупать» (волна — очередная атака врагов, игра идёт волнами). Или как должно быть: «Она должна выходить один раз и остаться там до начала боя». Причину оставляю Claude.

Почему работает — Симптом с цифрой Claude сводит с логами. Причину ищет тот, у кого больше данных. После серии таких отчётов я написал: «самый балансный билд, который у нас получался» (билд — собранная версия игры).

Где ломается — На скриншоте без логов: скриншот показывает симптом, причина обычно в логах и в выводе терминала — окна, где программы запускают текстовыми командами и куда они пишут ошибки. Правило «ошибку присылаю целиком» я пока не закрепил (правило 9 в главе 5).

Ступень — CLAUDE.md игры: «после каждого плейтеста разбирать его логи».

Попробуйте — Следующее замечание к работе AI напишите по шаблону 4 из главы 8.

15 Сомневаюсь в методике проверки

Ситуация — Claude приносит цифры: процент побед, прогноз, результаты тестов. Цифры правдоподобные, и их хочется принять.

Как делаю — Спрашиваю, как их получили: «а тесты были с учётом уровней сложности? если это были тесты на 1 сложности без ступеней, как будто мб такие винрейты и ок» (винрейт — доля побед).

Почему работает — Цифру можно посчитать верно в кривых условиях. В игре все прежние тесты оказались на нулевой ступени сложности. Вторая находка той же игры: боты-тестеры сильно переоценили живого игрока (цифры — в главе 6).

Где ломается — Если спрашивать только про итог. На «всё работает?» Claude ответит «тесты зелёные», и это будет правдой. Чего тесты не видят — в ошибке 1.

Ступень — Правило в памяти и в CLAUDE.md игры: баланс сначала считаем формулами и таблицей, ботов запускаем только для сверки.

Попробуйте — Получив от AI цифру, спросите: «На каких данных и в каких условиях ты это получил?»

16 Возвращаю пункт отчёта

Ситуация — Claude отчитался длинным списком: всё сделано. Перечитывать всё долго, верить на слово рискованно.

Как делаю — Беру один пункт и возвращаю с вопросом «закрыто?»: «Всё кайф, но не увидел онбординга чет, ты его сделал?» Онбординг — вводные экраны, которые учат пользоваться продуктом. Перед новым этапом прошу закрыть хвосты: «однако перед этим давай пофиксим всё то, что ты пометил как отложенное».

Почему работает — Выборочная проверка ловит недоделки без перечитывания всего. Тот онбординг не попал в рабочую версию: истёк ключ доступа для выкладки. Ключ доступа — пароль, по которому программа входит в сервис. В документе для заказчика возвращённый пункт оказался поправлен наполовину.

Где ломается

Когда статус тонет в длинном отчёте (пример — в приёме 9). Спасают нумерованные вопросы и строка статуса в конце. Это правило я тоже пока не закрепил.

Ступень

Привычка.

Попробуйте

Попросите в конце каждого отчёта строку «сделано / не сделано / ждёт меня». Раз в отчёт возвращайте один пункт с вопросом «закрыто?».

Закрепление

17 Превращаю правку в принцип

Ситуация

Claude ошибся, вы поправили. Через день та же ошибка всплывает в другом месте.

Как делаю

К точечной правке добавляю принцип: почему так и где ещё это применить. «Давай без значка: хочется, чтобы интерфейс визуально показывал логику и статус без необходимости дублировать текстом…» Короткие правила сразу распространяю на весь проект: «Для предлогов добавь неразрывные пробелы везде».

Почему работает

Одна правка закрывает целый класс ошибок. Принцип про значок стал правилом в памяти «статус показывает картинка», и Claude применяет его без напоминаний. Всего правил в памяти 9, у каждого есть «почему» и дата (см. принцип 7).

Где ломается

Правило на весь проект доходит не до всех мест. Продукт я переименовал одной фразой: «Давай сразу забудем про чёлку и будем жить в контексте, что это по дефолту усы». «Чёлка» и «усы» — старое и новое название продукта, чёлкой зовут вырез вверху экрана MacBook. «По дефолту» — по умолчанию. Старое имя после этого осталось в 18 строках сайта. При переименовании нужны поиск по всем файлам проекта и список мест, где осталось старое имя.

Ступень

Правило в памяти. Туда идут правила о том, как я работаю вообще. Правило одного продукта — в его CLAUDE.md (лестница — в главе 2).

Попробуйте

В следующей правке допишите «и везде, где…». Потом попросите: «Запиши это правилом: формулировка, почему, как применять».

18 Перед паузой делаю сводку и файл статуса

Ситуация

Работу надо прервать: выключить компьютер, отложить проект, перенести проекты на другой аккаунт. Всё, что живёт только в чате, может потеряться.

Как делаю

Прошу сводку и файл статуса и только потом останавливаюсь: «давай пока саммари сделаем и стопнем, т.к. комп надо выключить, потом продолжим. Сможем так без потери инфы?» Перед переносом проектов на другой аккаунт попросил шпаргалку, что где лежит.

Почему работает	Через сутки я вернулся одним словом «продолжай» и ничего не потерял. Перенос прошёл по шпаргалке и скрипту — небольшой программе, которая сама выполняет цепочку действий. Перенеслись 7 проектов из 7. Файлы STATUS.md, docs/next.md и CLAUDE.md работают как упакованное состояние проекта.
Где ломается	Если вернуться без адреса: без названия, пути и доступа Claude сначала ищет контекст (ошибка 8). Нужны название проекта, путь и что подключено, или одна фраза «Читай docs/next.md».
Ступень	CLAUDE.md и файлы статуса в проекте.
Попробуйте	Перед следующей паузой попросите записать сводку в файл. Вернувшись, проверьте, хватит ли одного слова «продолжай».

Лайфхаки

Мелкие приёмы, которые пригодятся в любой работе с AI.

— **Голосовой бриф по порядку.** Длинную задачу надиктовываю голосом, но держу порядок: проблема → идея → результат → формат. Свои идеи помечаю как гипотезы: «Это мои мысли, их нужно будет провалидировать...» После самоисправлений вроде «хотя нет, давай кавычки оставим» проверьте, что Claude взял последнюю версию.

— **Скрин → что жать → скрин.** Для интерфейса, где ошибка дорогая: «Простую стратегию: я буду присылать скрины, ты будешь говорить, что выбирать из реролов» (реролы — новые случайные наборы вариантов в игре). Так прошло около 55 скриншотов подряд. Старый скриншот сбивает Claude с толку, шлите только свежий.

— **Флаг «старт с 20-й волны».** Флаг — переключатель в коде: игра стартует сразу с 20-й волны, и финал проверяется за пару минут. Для курса аналог — тестовый вход сразу в последний модуль.

— **Нарочно слабый вариант.** Прошу собрать и вариант, в который сам не верю: «я уверен, что он плохой, но для согласования надо сравнить».

— **Проба на малые деньги до подписки.** Перед оплатой сервиса генерации — тест: 13 картинок за 47 ₽. Дальше у каждой пробы была названа цена.

— **«Я проверяю параллельно».** Смотрю результат, пока Claude ещё работает: «Если уже реализовал правки, о которых я говорю, то напиши (я просто сразу же чекаю и сразу же пишу...)». Так одна правка не делается дважды.

— **Правки по номерам скриншотов.** «скрин 4 — дизайн поехал». Claude отчитывается теми же номерами: «Все 7 правок сделаны».

— **Нет слов — аналогия.** «смесь Cataclismo и 9kings»: две чужие игры объясняют мою быстрее абзаца. Застрял — так и пишу: «а то чёт я совсем stuck».

— **На ошибку — симптом без претензий.** «обновил, пока всё так же» и скриншот. Или «мне чёт не нравится композиционно, без негатива».

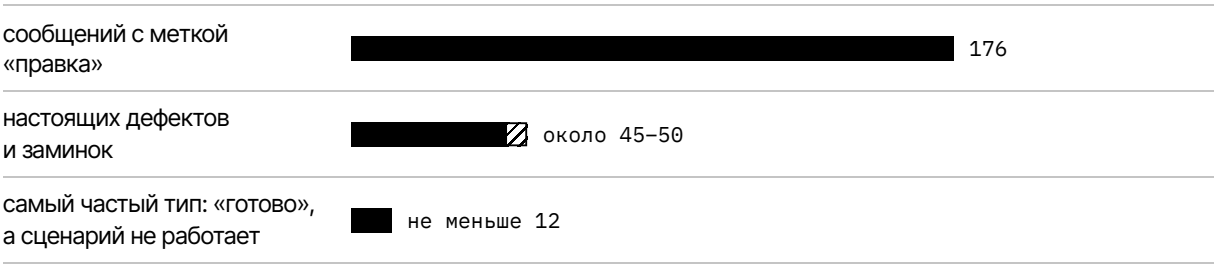
— **Запасной вариант прямо в постановке.** Разрешите Claude отложить сложное, если не выходит. Как это сработало в Barba, моём приложении для MacBook, — в главе 6.

5. Ошибки и правила, которые из них выросли

Чаще всего AI ошибается в слове «готово». Мои ошибки — в том, чего я не сказал в брифе. Лечатся и те и другие одинаково: повторившаяся ошибка становится правилом и поднимается по лестнице закрепления (глава 2).

Скилл ai-playbook нашёл в переписке 176 сообщений, которые он автоматически пометил как «правку». Настоящих дефектов и заминок среди них около 45–50. Остальное — новые задачи, вердикты по плейтестам, вставленные тексты и сбои сервиса.

Сколько «правок» оказались ошибками



Большинство «правок» — обычная работа. Среди настоящих ошибок чаще всего подводит слово «готово».

Ниже девять ошибок: пять у модели и четыре у меня. У каждой — симптом, причина, правило и его статус. **Закреплено** — правило есть в памяти, скилле или CLAUDE.md. **Локально** — в файлах одного-двух проектов. **Не закреплено** — живёт только в переписке.

Ошибки модели

Ошибка 1

«Готово», а сценарий не работает

Симптом — Claude отчитывается: «выложено, тесты зелёные». А кнопка не реагирует. Или вёрстка: «Все слиплось в кучу, абсолютно не читаемо». Так было в пяти проектах.

Причина — Автотесты проверяют данные. Клик по кнопке и вёрстку на экранах разной ширины никто не смотрит.

Правило — «Готово» — только после сценария: клик по каждой новой кнопке, скриншот каждого нового экрана, проверка на узком и широком экране. Что не проверено, Claude называет.

Статус — Локально. В CLAUDE.md игры стоит «НАБЛЮДАЕТСЯ или НЕ ПРОВЕРЕНО», в двух других проектах — проверки скриптами. В памяти общего правила нет, и это главный кандидат туда (принцип 5).

Ошибка 2

Раздувание: агенты, токены, текст

Симптом	Ревью сразу несколькими агентами съедало 2,3–2,8 млн токенов, ревью одним — 0,3 млн при сопоставимом качестве. Текст, переписанный моим скиллом стиля, «стал раза в 3 длиннее».
Причина	По умолчанию работа шла вширь. Глубокий режим рассуждений стоял у 72% ответов. Агентов я разрешал охотно, бюджет им не задал ни в одном старте.
Правило	Точечную правку Claude делает сам, без агентов и за один заход. Глубокий режим — только по моему слову. Режим и модель Claude объявляет строкой, расход прикидывает до прогона. Тексты не раздуваются.
Статус	Закреплено: два правила в памяти, CLAUDE.md рабочего проекта и скилл стиля (принцип 6).

Ошибка 3

Ломает рабочее при правке

Симптом	Правка баланса сделала игру тривиальной: «к 20-й волне было 2000 свободного золота». В другом продукте главное ограничение «админ не видит, кто что выбрал» нарушалось дважды.
Причина	Ограничение, сказанное один раз, теряется при следующих фичах. В проверки оно не превращено.
Правило	Баланс — формулами и таблицей, ботов гонять коротко, сложность задавать по живому игроку. Каждая правка — локальный коммит, чтобы было куда откатиться. На сервер правки уходят пачкой и по моему «да». Главное ограничение продукта проверяется после каждой фичи.
Статус	Баланс — закреплено, правилом в памяти. Коммиты и проверка ограничений — не закреплено.

Ошибка 4

Неверно читает референс

Симптом	В промпте генератору картинок стояло «как классическая Зельда» (Зельда — известная серия игр), и генератор нарисовал героя Зельды: «чет вообще не то получилось».
Причина	Референс пришёл без строки, что из него брать. Известное имя генератор понимает буквально.
Правило	К референсу — строка «берём это, не берём то». В промптах генераторов нет имён известных персонажей. Картинку-референс дублировать словами.
Статус	Не закреплено. Ближе всего к нему — разбор образца по трём корзинам (приём 6).

Ошибка 5

Говорит сложнее, чем я понимаю

Симптом	Про надпись в игре: «я не понял, что это значит: молчат защитники». Про черновик поста: «ты написал то, что я, перечитав 10 раз, не понял, поэтому я переписал так, как понял». Мой вариант поста набрал 748 показов при медиане моих постов около 300.
Причина	Уровень читателя в брифе не задан. Стоит его назвать, и Claude сразу перестраивает объяснение.
Правило	Глаголы буквальные. Статус в интерфейсе показывает картинка. Черновик не умнее автора. Логика проговаривается вслух.
Статус	Закреплено: два правила в памяти и руководство по моему стилю письма.

Мои ошибки

Ошибка 6

Мобильная версия и онбординг — только после живого теста

Симптом	Телефон в брифе был назван: «запускается в вебе с телефона», то есть в браузере телефона, без установки приложения. И всё равно мобильную версию почти каждый раз приходилось просить отдельно. Онбординг во всех трёх продуктах для вечеринок появился только после живого теста. Друзья не понимали продукт: «Ток я не поняла: я нажала, что скинусь, а дальше че?»
Причина	Среда и понятность не стали критериями готовности первого брифа.
Правило	В первом брифе — «проверь на узком экране» и «онбординг входит в MVP», то есть в первую рабочую версию. До рассылки — тест на двух людях.
Статус	Не закреплено. Но порядок уже другой: в октябрьской игре сначала были серые болванчики и проверка понятности, графика — потом. Серые болванчики — простые серые фигуры на месте будущей графики.

Ошибка 7

Секреты в чате

Симптом	Ключи доступа к GitHub — минимум четыре раза. GitHub — сайт, где хранят код и историю его версий. Ещё ключи внешних сервисов, root-доступ к серверу, то есть полный доступ ко всему на нём, и ссылка с ключом внутри.
Причина	Вставить ключ в сообщение — самый короткий путь к доступу. Потом ключ остаётся в истории сессии.

Правило — Ключи в сообщения не пишутся. У меня сработали три способа:

— вход в GitHub по одноразовому коду, с хранением в Keychain — связке ключей macOS;

— ключ в файле вне репозитория, то есть вне папки с кодом;

— перевыпуск ключа, попавшего в чат.

Статус — Не закреплено. Скрипт, который собирал этот гайд, маскирует секреты при выгрузке. В самих сессиях они остались.

Ошибка 8

Продолжение без адреса

Симптом — Около 12 стартов вида «работа остановилась, давай продолжим» — без названия, пути и доступа. В пяти из них Claude сначала искал контекст или просил подключить папку.

Причина — Новая сессия начинается с пустого контекста. Где лежит прошлая работа, Claude не знает.

Правило — Для продолжения — название проекта, путь к папке и что подключено. Или одна фраза: «Читай docs/next.md».

Статус — Локально: docs/next.md есть в игре, общего правила нет (приём 18).

Ошибка 9

Одна сессия на всё

Симптом — Одна сессия на 129 сообщений и 72 дня вела шесть разных тем. В сессии игры — 159 сообщений и 18 сжатий контекста. Договорённости при этом теряются.

Причина — Состояние проекта жило в истории чата, и уйти из сессии значило его потерять. Длина бьёт по лимитам, обрывам и стоимости. Доля правок от неё не растёт: 16–23% в длинных сессиях, 41% у короткой вычитки.

Правило — Одна итерация — одна сессия, состояние — в файле, новая тема — новая сессия. Форк здесь не годится: он тащит за собой всю историю.

Статус — Закреплено для игры: правило в памяти, его осталось обобщить (приём 13).

Что я до сих пор не закрепил: проверку перед «готово», ошибку целиком и секреты — правила 1, 9 и 10 ниже. Первое и последнее — следующие кандидаты в память.

Десять правил

Выжимка из всех ошибок, которые нашёл скилл: правила 5, 7 и 8 выросли из случаев, которые в главу не вошли. Правила 9 и 10 — для меня, остальные обращены к Claude.

1 Не пиши «готово», пока не прошёл мой сценарий: клик, скриншот, узкий экран. Что не проверил — назови. *Локально, следующий шаг — в память.*

2 Точечную правку делай сам. Агентов и глубокое ревью — по моему слову, режим объявляй строкой, расход прикидывай до запуска. *Закреплено.*

3 Одна итерация — одна сессия, состояние — в файле. *Закреплено для игры, осталось обобщить.*

4 Числа сначала формулами и таблицей, сложность — по живому человеку. *Закреплено.*

5 Новое — только под проблему, которую я почувствовал в тесте. *Закреплено.*

6 Пиши буквально и на моём уровне; черновик не умнее автора. *Закреплено.*

7 У цифры и обещания есть источник; непроверенное помечай. *Локально, осталось обобщить на документы.*

8 Решения — с датой и условием пересмотра; перед правкой общего документа сверяй версию. *Локально.*

9 Ошибку присылаю целиком: вывод терминала, логи выкладки, скриншот с контекстом. *Не закреплено.*

10 Для секретов — только код входа, Keychain или файл вне репозитория. Засветился в чате — перевыпусти. *Не закреплено, следующий шаг — в память.*

Как превратить формулировку в правило с «почему» и датой — в главе 2 и в упражнении 3 главы 9.

Как я решаю на развилках

Из 75 рекомендаций Claude на развилках я принял 63 (принцип 3). Граница такая:

— **Принимаю,** когда вопрос в чужой экспертизе: «Я вообще не разбираюсь. Поэтому тут давай как лучше сделать». На выкладке продукта — 5 раз из 6.

— **Пишу своё,** когда в варианте спрятаны деньги и лимиты, риск потерять данные, лишняя сложность или выбор вслепую. На последний случай: «сделай пре-рендер двух вариков, и я выберу» (пре-рендер — заранее отрисованный пример, «варик» — вариант).

Попробуйте так же. В технике соглашайтесь с рекомендацией. На деньгах, данных и сложности остановитесь и решите сами.

---

6. Кейсы

Четыре истории: игра, свой продукт, маленький бот и AI-агент для курса. Схема одна: задача → как работал → результат → что переносимо. Номера приёмов — из главы 4.

«Скрижаль»: от сна до itch.io

Задача — Первая игра по концепту из сна: tower defense про суд. В этом жанре игрок строит оборону от волн врагов.

Как работал

— Сначала база (принцип 1): «Сначала заэкстракти в себя все бест-практисес по играм и геймдизайну… После я пришлю концепт игры… и обсудим». «Заэкстракти в себя» — собери и усвой, «бест-практисес» — лучшие практики. Получились свод практик на 18 тыс. слов и ревью концепта.

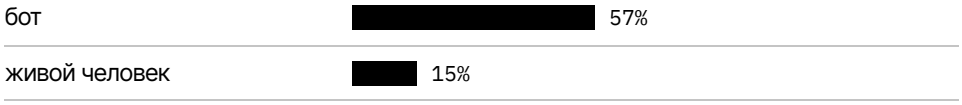
— Цикл плейтеста: собрал → поиграл → ощущение и цифра → Claude сводит с логами → правка (приём 14). Критерий сложности — объяснимый проигрыш (приём 3).

— Сомнение в методике проверки (приём 15).

Результат

По моей оценке, «самый балансный билд, который у нас получался». Публичная сборка на itch.io, площадке инди-игр: 7 ступеней сложности и финальный босс. Главный урок — расхождение бота и живого игрока.

Доля побед в «Скрижали»



Боты переоценивают человека, поэтому баланс считается математикой и проверяется на людях.

Отсюда правило в памяти «баланс считаем математикой» (приём 15, глава 2): числа сначала формулами, боты только для сверки.

Что переносимо

Симптом с цифрой после каждого плейтеста. Утверждённые настройки баланса меняются только по вердикту человека.

Barba: от вечера до релиза

Задача

Свой продукт по мотивам рилса — короткого видео в Instagram: приложение в вырезе экрана MacBook.

Как работал

— Сначала рисёрч, потом MVP по нумерованному списку из шести функций. И просьба «прочелленджи и провалидируй идею» (приём 8).

— Запасной вариант прямо в постановке: «если нереально — давай пока клод скипнем» («скипнем» — пропустим). Через сутки отложенное стало фичей.

— Проверка руками (принцип 5): жест и системное уведомление Claude сам не проверит. Я проверял параллельно, не дожидаясь конца разработки.

— Перед промо-роликом спросил: «у меня ток 17% недельных лимитов осталось, сколько примерно наша работа займёт?» Claude предложил экономный план: результат смотрю я, он сверяет кадры точечно (принцип 6).

Результат — MVP собран за 22 минуты, с 18:57 до 19:19. Дальше около 30 итераций за две недели до релиза 0.3.9.

Что переносимо — Запасной вариант прямо в постановке. План под остаток лимитов.

Бот погоды за 12 минут

Задача — Telegram-бот, который присылает погоду из места, где хуже.

Как работал — Одно жёсткое ограничение «чтобы он не ел денег» и пример поведения (приём 1).

Результат — Ограничение решило архитектуру: бот работает без нейросетей, на бесплатном сервисе погоды. От брифа до работающего бота прошло около 12 минут, с 18:25 до 18:37. После моей короткой проверки ввод города исправили за один заход.

Что переносимо — Ограничение отсекает лишние решения до начала сборки. Ищите то, которое меняет устройство продукта: деньги, приватность, кто что делает.

AI-агент для курса: готово, когда тесты зелёные без меня

Задача — AI-агент, который превращает сценарии уроков в пакет курса, готовый к загрузке на платформу. Проект для заказчика, поэтому без подробностей. Главный вопрос кейса касается любого агента: как принимать его работу, не проверяя каждый результат сам.

Как работал —

- Вопросы к созвону, потом расшифровки двух записей встречи (приём 5). Из них всплыли факты, которых не было в ТЗ.
- План первой версии дала цепочка: ревью, поиск аналогов, разбор задач, пограничные случаи, три варианта архитектуры, судьи и критик (принцип 1). Судьи — агенты, которые сравнивают варианты, критик — агент, который ищет в них слабые места.
- В самом продукте два агента. Создатель собирает из сценариев уроков текст курса, загрузчик упаковывает его в архив для платформы. Выход создателя я задал под вход загрузчика: текст, «который и проверить адекватно можно, и который при этом наш другой агент съест и переделает в зип». Зип — архив, один файл со всеми материалами курса. Так появился контракт «создатель ⇔ загрузчик» — договорённость о формате между двумя агентами: что один отдаёт и в каком виде, чтобы другой это принял.
- Критерий готовности (принцип 5): «давай автотесты напишем и прогоним агента до тех пор, пока все тесты не будут нормально отрабатывать = агент всё делает хорошо, моё вмешательство не нужно». Автотесты (глава 2) проверяют результат каждый раз одинаково. Красный тест значит, что проверка провалилась.

Для методиста

Логика знакомая: тесты для агента — те же критерии оценивания для ученика: их пишут заранее, их можно проверить, и они одинаковы для каждой попытки.

Результат

Проверка сложилась из трёх слоёв:

— 689 модульных тестов. Модульный тест проверяет одну часть агента.

— 37 тестов загрузчика. Они проверяют, что архив для платформы собирается правильно.

— Прогоны по кейсам. Кейс здесь — тестовый пример: сценарий урока и то, что агент должен из него собрать. Кейсов 61, результат агента по каждому сравнивают с ожидаемым. Прогонов было четыре: третий прошёл 57 из 61, провалы починили.

Модульные тесты и тесты загрузчика зелёные. Приёмка — «два зелёных подряд»: два прогона подряд, в которых прошли все тесты. Один зелёный прогон может оказаться удачей. Два подряд показывают, что агент работает стабильно.

Есть проверка ещё проще — смоук-тест. Это быстрая проверка главного сценария: запускается ли продукт вообще. Название пришло от проверки приборов: включили и смотрят, не пошёл ли дым. В CLAUDE.md моей игры смоук-тест стоит в паре со скриншотом (ошибка 1). Как настроить такие тесты для своего агента — в следующем, более глубоком гайде.

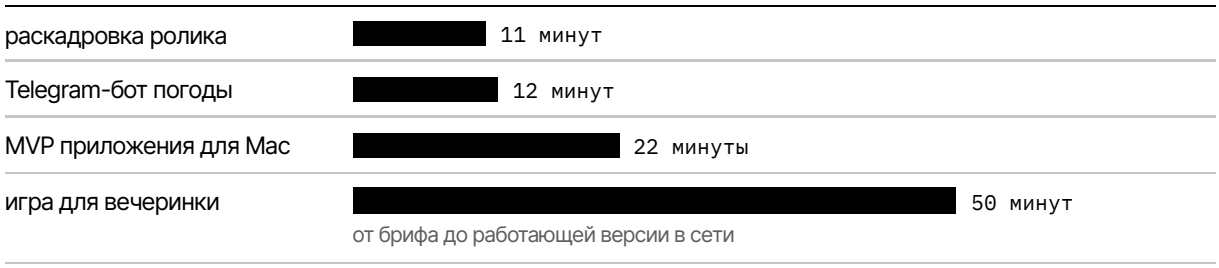
Цена и урок

«Дешёвый» прогон съел около 10 млн токенов. С тех пор расход прикидывается до прогона (принцип 6).

Что переносимо

Выход одного агента проектируется под вход следующего. Критерий «без моего вмешательства» превращается в тесты, приёмка — в два зелёных прогона подряд. Так можно принимать работу любого агента.

Скорость первой версии



Собрать первую версию — дело минут. Дольше всего — сделать её понятной и проверить на людях.

Все четыре первые версии собраны меньше чем за час. Дальше узкое место — понятность с первого раза, и проверить её могут только люди. Во всех трёх продуктах для вечеринок онбординг появился после живого теста (ошибка 6). Сэкономленное на сборке время вкладывайте в стоп-точки до неё (приём 4) и в проверку людьми после (принцип 5).

7. Инструменты

Инструмент я выбираю по двум вопросам: что нужно сделать и сколько я готов на это потратить.

Инструмент под задачу

— **Claude Code** — там, где нужны терминал, Git и деплой. Git — программа, которая хранит историю версий кода. Команды Claude Code запускает сам. В апреле я делал такие задачи в Cowork и собрал около 15 сбоев среды — случаев, когда Claude не хватало нужной программы, прав или доступа. С июля они ушли в Claude Code, и сбои почти исчезли. Если что-то сломалось, присылайте ошибку целиком: вывод терминала и логи выкладки (правило 9 в главе 5).

— **Облачная и локальная сессии.** Облачной отдаю бумагу, локальной на маке — интерфейс и графику (приём 13).

— **Claude как штурман.** Картинки, дизайн и прототипы делают Claude Design, генераторы картинок и Lovable. Claude пишет для них промпт, я генерирую и приношу результат обратно (приём 11, шаблон 5).

Бюджет тоже часть задачи: под остаток лимитов Claude может предложить экономный план (Barba, глава 6).

Режимы

Режим — это модель (подешевле или подороже) и глубина её работы (быстрая или глубокая). За полгода я 26 раз переключал модель вручную, пока не попросил Claude выбирать режим самому (принцип 6).

Теперь в памяти лежит правило: быстрый режим по умолчанию, глубокий — по моему слову. Разговор и дизайн Claude ведёт сам, механическую работу отдаёт модели подешевле. Режим он объявляет строкой перед работой, и я могу ответить «глубже» или «дешевле».

Где хранить договорённости

Договорённости из чата теряются при сжатии контекста (глава 1). Поэтому я держу их в файлах — на ступенях лестницы из главы 2.

Память. Сюда идут правила о том, как вы работаете вообще. Правило одного продукта — в инструкцию проекта. У меня в памяти 9 правил, у каждого есть «почему» и дата. Например, «отвечать по-русски» лежит там с 30.09. В разборе стартов нашлось шесть первых ответов на английском при русском запросе.

CLAUDE.md. Сюда идут правила одного проекта: как проверять, что считать готовым, где что лежит. У меня CLAUDE.md в 7 проектах. Для продолжений рядом лежит docs/next.md, и новая сессия стартует с фразы «Читай docs/next.md» (приём 18).

Скилл. Нужен, когда действие из нескольких шагов повторяется. У меня их 17: 7 личных и 10 в проектах. Например, скилл better-text пишет в стиле моих лонгридов.

Агенты и субагенты. Субагентов Claude запускает на рисёрч, ревью или проверку, можно несколько параллельно. В моих сессиях их было 93, максимум — 18 в одной. Они дают ширину: много источников, ревью с разных сторон. Точечную правку Claude делает сам, за один заход. Сколько агентов можно запускать, пишите в брифе (ошибка 2). Агенты на четвёртой ступени лестницы — такие помощники, записанные в файл, чтобы запускать их снова. У меня их 30, и все — боты-тестеры игры: они проходят уровни вместо живых игроков. Как они переоценили человека — в главе 6.

Задача по расписанию. Нужна для рутины по времени: регулярного поиска, мониторинга. У меня 5 таких задач и 63 запуска.

Чего Claude не проверит сам

— Жесты и системные уведомления. В Barba их проверял я, руками (глава 6).

— Видео целиком «на глаз»: ролик смотрю я.

— Ощущение от игры: боты-тестеры его не заменят (глава 6).

— Понятность для нового человека (ошибка 6).

— Вкус. В CLAUDE.md моей игры так и записано: «Вкус решает Даниил».

Поэтому просите Claude назвать, что он не проверил. И проверяйте это сами там, где результатом будут пользоваться (принцип 5).

---

8. Шпаргалка

Чек-лист брифа

Скилл разобрал 50 стартов моих сессий. Старты, после которых Claude сразу шёл по пунктам, собраны из восьми элементов:

1. **Зачем.** «Цель —...», «чтобы...».
2. **Что было до.** Прошлая работа с названием и путём к файлам (ошибка 8), что на скриншотах.
3. **Что сделать по этапам.** «Сначала..., после...» или список 1) 2) 3).
4. **Чего нельзя.** Одно жёсткое ограничение или антипример (приёмы 1 и 2).
5. **В каком виде.** Файл, блоки, для кого.
6. **Критерий готовности.** Что должен увидеть или понять человек (приём 3).
7. **Право спорить.** «я ок с критикой», «проверяй меня», «прочелленджи».
8. **Что делаю я.** «текст напишу сам», «концепт пришлю после».

Позицию я задаю глаголом: «отревьюй», «провалидируй», «критически оцени». Роль вроде «ты — эксперт...» в моих стартах не встретилась ни разу.

Как часто элементы брифа были в моих 50 стартах

Зачем	около 30 из 50
Что сделать по этапам	около 8 из 50
Право спорить	около 8 из 50
Критерий готовности	около 6 из 50

Чаще всего в моих брифах не хватало критерия готовности.

Остальные элементы скилл не считал поштучно. «Что было до» и «чего нельзя» встречались часто, но около 12 стартов ссылались на прошлую работу без названия и пути. «В каком виде» — редко, «что делаю я» — несколько раз.

Строки по умолчанию

Эти строки — выводы из ошибок. Первая лежит у меня в памяти, остальных в моих брифах пока нет.

— «Отвечай по-русски».

— «Готово, когда <критерий>».

— «Проверь на узком экране» — для всего, что открывают с телефона.

— «Проект лежит в <путь>, доступ подключён» — для продолжений.

— «Бюджет: не больше агентов, глубокий режим только по моему слову, прикинь расход до запуска».

— «В конце отчёта одна строка: сделано / не сделано / ждёт меня».

Чек-лист проверки результата

Пройдите его сами или попросите Claude пройти до слова «готово».

— Кликнуть по каждой новой кнопке.

— Посмотреть скриншот каждого нового экрана.

— Открыть на узком и широком экране.

— Спросить: «Что не проверено?»

— Выборочно вернуть пункт отчёта: «Это закрыто?» (приём 16).

— Проверить инварианты продукта (приём 1).

— В документах: у каждой цифры есть источник, непроверенное помечено.

Пять шаблонов

Шаблоны собраны из моих сообщений: конкретика заменена на <…>, примеры — из работы методиста. Остальные 12 шаблонов будут в следующем, более глубоком гайде.

Шаблон 1

Продукт со стоп-точками

Для нового продукта или большой задачи с нуля (приём 4).

```
Контекст: <что за продукт, для кого и где им будут пользоваться — например, «тренажёр,
ученики проходят его с телефона»>.
Материалы: <ТЗ, программа или план, которые уже есть>. Я ок с критикой.
Главное для меня — <критерий успеха: что должен почувствовать или понять пользователь>.

1) Посмотри, оцени, провалидируй.
2) Составь план: несколько версий MVP с точками моего контроля и чек-листом — на что
мне смотреть, где хватит короткого фидбека, где нужен подробный.
3) Задай мне вопросы.
4) Реализуй.

Отвечай по-русски. В конце отчёта одна строка: сделано / не сделано / ждёт меня.
```

Последних двух фраз в моём оригинале не было: их подсказали строки по умолчанию.

Шаблон 2

Проверь гипотезу

Когда кажется, что вы нашли проблему, но уверенности нет (приём 8).

```
У меня гипотеза: <утверждение и почему я так думаю — например, «задания второго модуля
слишком сложные для новичков»>.
Прорисёрчи и провалидируй. Сначала анализ — чинить пока не надо.
Если я ошибаюсь — скажи прямо.
```

Шаблон 3

Решения от меня до правок

Когда Claude собрал список замечаний и готов чинить (приём 9).

```
Выпиши в чат решения, которые от меня нужны перед правками: с вариантами и твоей
рекомендацией. Вопросы пронумеруй.
```

С номерами можно ответить одним сообщением, как я: «1) а 2) а 3) в 4) в 5) а 6) в» (приём 10).

Шаблон 4

Отчёт после теста

Когда вы сами прошли урок, тренажёр или прототип (приём 14).

```
Что проверено: <что>. Цель проверки: <…>. Не проверено: <что>.
Ощущение: <сравнение с прошлой версией>.
Симптом: <одна цифра — например, сколько учеников дошли до конца урока>.
Должно быть: <ожидаемое поведение>. Сейчас: <фактическое>.
```

Шаблон 5

Промпт для другого инструмента

Когда нужна картинка, дизайн или прототип в другом сервисе (приём 11).

```
Проработай пока только концепцию <чего — например, обложки курса или экрана тренажёра>
и напиши промпт для <Claude Design / генератора картинок / Lovable>.
Что берём из референса: <…>. Что не берём: <…>.
Я сгенерирую и принесу результат — соберёшь лучшее из референсов.
Пока я генерирую, ты параллельно <другая задача>.
```

Строку про референс я добавил после ошибки 4.

---

9. Практика

Три упражнения на ваших задачах. На все уйдёт около 30 минут.

Упражнение 1

Перепишите последний промпт по каркасу

Возьмите промпт, после которого пришлось переделывать результат. Допишите недостающее по чек-листу брифа.

Готово, если в промпте есть «чтобы…», одно жёсткое ограничение и критерий готовности.

Упражнение 2

Поставьте стоп-точку

В следующей задаче попросите до правок: «Выпиши решения, которые нужны от меня» (шаблон 3). Или разбейте работу на этапы: оценка → план → вопросы ко мне → сборка (приём 4).

Готово, если переделок меньше, чем в похожей задаче без стоп-точки.

Упражнение 3

Закрепите одну ошибку

Найдите ошибку AI, которая повторилась дважды за неделю. Запишите правило: формулировка, «почему», «как применять». Попросите Claude положить его в память или в CLAUDE.md проекта (приём 17).

Готово, если в следующий раз AI применил правило без напоминания.

Приложение. Как собран этот гайд

Задача

Превратить полгода работы с Claude в методичку, где у каждого вывода есть доказательство.

Как работал

Я отдал переписку скиллу ai-playbook. Дальше данные прошли такой путь.

Путь данных: от переписки к гайду

1. **55 сессий в Claude Code и Cowork, 06.04–16.10.2026** — 955 моих сообщений
2. **Скрипт выгрузки маскирует секреты**
3. **12 порций переписки**
4. **6 агентов разбирают порции, ещё 2** — срезы: постановку задач и ошибки
5. **Свод: приёмы, ошибки, кейсы, шаблоны** — каждый вывод с цитатой
6. **Этот гайд**

У каждого вывода в гайде есть адрес в переписке.

Результат

Каталог приёмов с цитатами, разбор ошибок, кейсы, 17 шаблонов и скелет этого гайда. Все цифры в гайде взяты оттуда.

Ограничения

— Один человек, один AI-ассистент, полгода.

— Чатов из claude.ai — сайта с обычным чатом Claude в браузере — в выборке нет: для них нужен отдельный экспорт.

— Старые сессии Claude Code могли удалиться автоматически.

— Автоматические метки вроде «правка» вручную проверены только для ошибок.

— Связь «ошибка → правило» восстановлена по датам.

Что переносимо

Свою практику можно изучать как данные: по переписке, без опоры на воспоминания. Рецепт целиком будет в следующем, более глубоком гайде.

Словарь

Термины из гайда простыми словами. В скобках — где о них подробнее.

AI-агент — программа, которая сама ведёт задачу из многих шагов: читает файлы, пишет тексты и код, запускает команды. Мой агент — Claude.

Claude — AI-ассистент компании Anthropic, в гайде он работает как агент. Обычный чат с Claude открывают в браузере, на сайте claude.ai.

Claude Code — приложение, где Claude работает как агент с кодом и командами. Команды он запускает сам; задачи с терминалом, Git и деплоем я делаю здесь (глава 7).

Claude Design — инструмент Anthropic для макетов и прототипов. Промпт для него пишет Claude, генерирует человек (приём 11).

CLAUDE.md — файл с инструкциями одного проекта в его папке: правила, состояние, проверки. Claude Code читает его сам при старте каждой сессии (глава 2, глава 7).

Cowork — приложение, где Claude работает как агент с файлами и документами. Задачи с терминалом и Git там шли со сбоями (глава 1, глава 7).

Git и GitHub — Git хранит историю версий кода, к любой сохранённой версии можно откатиться. GitHub — сайт, где хранят код вместе с этой историей. Папку с кодом называют репозиторием (ошибка 7, глава 7).

Lovable — сервис, который собирает сайт или приложение по текстовому описанию (приём 11).

MVP — первая минимальная рабочая версия продукта (ошибка 6, глава 6).

автотест — небольшая программа, которая сама проверяет результат, каждый раз одинаково. Данные она проверяет, клик по кнопке и вёрстку на узком экране не видит. Модульный тест проверяет одну часть программы (глава 2, глава 6).

баланс — насколько игра трудная: сила врагов, цены, запас ресурсов. Его считают формулами и таблицей, ботами только сверяют (глава 2, приём 15).

бот — программа в мессенджере, которая сама принимает и присылает сообщения, например Telegram-бот погоды (приём 1, глава 6).

бот-тестер — программа, которая сама проходит игру вместо живого игрока. У меня 30 таких агентов, записанных в файлы. В «Скрижали» доля побед бота — 57%, у живого человека — 15% (глава 2, глава 6).

вайрфрейм — схема экрана: что где стоит, без оформления (приём 13).

генератор картинок — сервис, который рисует картинку по текстовому описанию. Промпт для него может написать Claude (принцип 5, приём 11).

«два зелёных подряд» — условие приёмки агента: два прогона подряд, в которых прошли все тесты. Один зелёный прогон может оказаться удачей, два подряд показывают, что агент работает стабильно (глава 6).

деплой — выкладка продукта в интернет, после которой им можно пользоваться. «Выложено» — промежуточный статус: готово, когда результат проверил человек (приём 4, принцип 5).

задача по расписанию — задание, которое Claude Code сам запускает в заданное время, без человека: регулярный поиск, мониторинг. Верхняя ступень лестницы закрепления (глава 2, глава 7).

зелёный тест — проверка, которая прошла без ошибок; красный — проверка, которая провалилась. «Тесты зелёные» значит, что прошли все автотесты (принцип 5, глава 6).

инвариант — свойство продукта, которое не должна ломать ни одна правка, например «админ не видит, кто что выбрал». Его проверяют после каждой доработки (приём 1, ошибка 3).

ключ доступа — пароль, по которому программа входит в сервис. Ключи и пароли называют секретами: в чат их не пишут (приём 16, ошибка 7).

коммит — сохранённая версия кода. Локальный коммит остаётся на вашем компьютере, push отправляет версии на сервер (приём 12).

контекст — всё, что Claude держит в голове в текущей сессии: сообщения, файлы, результаты. В документации — context window (глава 1).

контракт между агентами — договорённость о формате: что один агент отдаёт и в каком виде, чтобы следующий это принял (глава 6).

контур — место, где идёт работа. Облачная сессия идёт на сервере Anthropic, без доступа к вашему компьютеру: ей отдают «бумагу» — тексты и схемы экранов. Локальная идёт на вашем компьютере и видит его файлы и программы: ей отдают интерфейс и графику (приём 13).

лестница закрепления — путь, по которому повторяющаяся правка становится правилом, которое AI применяет сам: разовый промпт → правило в памяти → инструкция проекта (CLAUDE.md) → скилл или агент → задача по расписанию. Ступень в карточке приёма показывает, докуда он дошёл (глава 2, глава 4).

лимиты — объём работы, который даёт подписка, например на неделю. Его считают в токенах (глава 1, глава 6).

логи — записи о том, что делала программа и где она сбилась. Скриншот показывает симптом, причина обычно в логах (глава 2, приём 14).

мокап — макет, на котором видно, как будет выглядеть результат. Выбор по мокапу стоит проверить на настоящих экранах (приём 10).

модель — AI-«движок», который читает и пишет текст. Бывают модели подешевле и подороже: механическую работу Claude отдаёт той, что подешевле (глава 7).

`НАБЛЮДАЕТСЯ / НЕ ПРОВЕРЕНО` — пометки из CLAUDE.md моей игры: ими Claude пишет в отчёте, видел ли он результат сам или не проверял его. Вкус всё равно решает человек (глава 2, ошибка 1).

онбординг — вводные экраны, которые учат пользоваться продуктом (приём 16, ошибка 6).

память — заметки-правила, которые Claude сам подгружает в каждую новую сессию. Сюда идут правила о том, как вы работаете вообще; у каждого есть «почему» и дата (глава 2, глава 7).

плейтест — проверка игры живыми людьми. Цикл плейтеста: собрал → поиграл → ощущение и цифра → Claude сводит с логами → правка (глава 2, глава 6).

провалидировать — проверить, выдерживает ли идея критику (приём 4, шаблоны 1 и 2).

прогон — один запуск задачи или проверки от начала до конца. Каждый расходует токены: однажды «дешёвый» прогон съел около 10 млн (глава 1, глава 6).

промпт — текст задания для AI. Разовый промпт живёт до конца сессии или до первого сжатия контекста, это нижняя ступень лестницы закрепления (глава 2).

путь — адрес папки проекта на компьютере. Без него Claude в новой сессии сначала ищет, где лежит прошлая работа (глава 2, ошибка 8).

развилка — момент, когда Claude показывает варианты решения и ждёт выбора человека. Из 75 рекомендаций Claude на развилках я принял 63 (глава 2, принцип 3).

режим — какая модель работает (подешевле или подороже) и насколько глубоко она рассуждает. Самый глубокий и дорогой — ультра. У меня по умолчанию быстрый режим, глубокий — по моему слову (принцип 6, глава 7).

референс — образец, который нравится и на который ориентируются: лендинг, курс, ролик (приём 6, ошибка 4).

сервер — компьютер в интернете, который работает круглосуточно. На сервер отправляют код, на нём работают боты и облачные сессии Claude (приёмы 1, 12, 13).

сессия — один непрерывный рабочий разговор с Claude, со своей историей и своим контекстом. Первое сообщение сессии — старт, в нём ставят задачу (глава 1, глава 2).

сжатие контекста — когда контекст переполняется, Claude сжимает его в краткий пересказ, и детали выпадают. В документации — compaction. Правила в файлах сжатие переживают (глава 1, принцип 7).

скилл — записанная в файл инструкция для повторяющейся задачи из нескольких шагов. Её запускают командой, или Claude подключает её сам; этот гайд собрал скилл ai-playbook (глава 7, приложение).

скрипт — небольшая программа, которая сама выполняет цепочку действий: например, переносит проекты на другой аккаунт или выгружает переписку (приём 18, приложение).

слоп — шаблонный AI-контент, который узнают с первого взгляда (приём 2).

смоук-тест — быстрая проверка главного сценария: запускается ли продукт вообще. С неё начинают, подробные тесты идут потом (глава 6).

спрайт — картинка в игре, например персонаж или предмет (принцип 5).

стоп-точка — место в задаче, где AI останавливается и ждёт решения человека. Мой порядок: оценка → план → вопросы ко мне → сборка (приём 4).

субагент — помощник, которого Claude запускает внутри сессии на часть работы: рисёрч, ревью, проверку. Работает в своём контексте, возвращает только итог, может идти параллельно с другими. В гайде их зовут просто агентами (глава 2, глава 7).

сценарий — путь пользователя в продукте: что он нажимает и что должен получить. «Готово» можно писать только после того, как сценарий прошёл (глава 2, ошибка 1).

терминал — окно, где программы запускают текстовыми командами и куда они пишут ошибки. Вывод терминала при сбое присылайте целиком (приём 14, глава 5).

токены — кусочки текста, которые модель читает и пишет. По ним считают лимиты и оплату: ревью несколькими агентами съедало 2,3–2,8 млн токенов, одним — 0,3 млн (глава 1, принцип 6).

файл статуса — файл в папке проекта, где записаны состояние дел и следующий шаг. У меня это docs/next.md и STATUS.md: новую сессию я начинаю с фразы «Читай docs/next.md» (приём 13, приём 18).

фича — функция продукта. Кор-фичи — главные функции (приём 2).

флаг — переключатель в коде, который меняет поведение программы: например, игра стартует сразу с 20-й волны (лайфхаки в главе 4).

форк — ответвление чата: копия со всей историей, от которой разговор идёт отдельно. Для новой темы лучше чистая сессия, форк тащит за собой всё старое (приём 13, ошибка 9).

ширина — работа вширь: сразу много агентов и параллельных прогонов. Даёт много источников и ревью с разных сторон и сильно поднимает расход (глава 1, принцип 6).

Что почитать

Здесь только официальная документация Anthropic. Все ссылки я проверил 16 октября 2026 года. Инструменты обновляются быстро: если интерфейс выглядит иначе, чем в гайде, верьте документации.

Каждая страница есть на русском и на английском. Русский перевод местами машинный, и часть терминов осталась по-английски: skills (скиллы), subagents (субагенты), context window (контекст). Где оригинал читается заметно лучше, первой стоит английская ссылка.

С чего начать:

— «Лучшие практики для Claude Code». Как ставить задачу AI-агенту. Два главных совета перекликаются с принципами 1 и 5: сначала исследовать и планировать, потом делать; дать Claude способ проверить свою работу. Раздел о том, как Claude берёт у вас интервью, близок к стоп-точкам из приёма 4. Страница написана для кода, но почти всё переносится на тексты и курсы. Перевод неровный, лучше читать оригинал:
  - en code.claude.com/docs/en/best-practices
  - ru code.claude.com/docs/ru/best-practices

— «Обзор инженерии подсказок». Входная страница руководства по промптам. Первый совет: прежде чем улучшать промпт, определите критерий успеха и способ его проверить. Это то же «готово, когда…» из чек-листа брифа в главе 8. Отсюда ведут ссылки на полный справочник по промптам и на интерактивный учебник:
  - ru platform.claude.com/docs/ru/build-with-claude/prompt-engineering/overview
  - en platform.claude.com/docs/en/build-with-claude/prompt-engineering/overview

Лестница закрепления из главы 2, по ступеням:

— «Как Claude запоминает ваш проект». Две памяти Claude Code. CLAUDE.md — файл с постоянными инструкциями проекта: вы пишете его сами, а Claude читает его в начале каждой сессии. Auto memory — заметки, которые Claude ведёт сам по вашим исправлениям и предпочтениям. На странице — где хранить эти файлы, как писать инструкции, чтобы их соблюдали, и как посмотреть и поправить память командой /memory. Пригодится к принципу 7:
  - ru code.claude.com/docs/ru/memory
  - en code.claude.com/docs/en/memory

— «Расширение Claude с помощью skills». Как устроен скилл в Claude Code и как сделать свой: папка с инструкцией, иногда со скриптами и файлами. Claude сам подключает скилл, когда задача подходит, или вы запускаете его командой:
  - ru code.claude.com/docs/ru/skills
  - en code.claude.com/docs/en/skills

— «Создание пользовательских subagents». Субагент — помощник, который работает в своём контексте, отдельно от основного разговора. Он берёт побочную работу (поиск, логи, чтение файлов) и возвращает в основной разговор только итог. Своего субагента стоит завести, когда вы раз за разом запускаете одного и того же исполнителя с одними и теми же инструкциями. Учтите: субагенты расходуют те же лимиты, что и основной разговор. Сколько это стоит на практике — в принципе 6:

  ru code.claude.com/docs/ru/sub-agents

  en code.claude.com/docs/en/sub-agents

— «Планирование повторяющихся задач в Claude Code Desktop». Верхняя ступень лестницы: задача, которую Claude Code сам запускает по расписанию. В начале страницы — таблица трёх способов: в облаке, на вашем компьютере и повтор внутри открытой сессии:

  ru code.claude.com/docs/ru/desktop-scheduled-tasks

  en code.claude.com/docs/en/desktop-scheduled-tasks

— «Расширение Claude Code». Карта, которая сводит все ступени вместе. Самое полезное на ней — таблица о том, как наращивать настройку со временем. Claude дважды ошибся в одном соглашении — запишите его в CLAUDE.md. В третий раз вставляете в чат одну и ту же процедуру — оформите скилл. Побочная задача забивает разговор — отдайте её субагенту. Получается та же лестница закрепления. В переводе смешаны «ты» и «вы», оригинал читается лучше:

  en code.claude.com/docs/en/features-overview

  ru code.claude.com/docs/ru/features-overview

Словарь:

— «Глоссарий». Короткие определения терминов Claude Code: context window (контекст), compaction (сжатие контекста), CLAUDE.md, skill, subagent. Статья про сжатие объясняет, зачем записывать уроки в файл: CLAUDE.md проекта и автоматическая память переживают сжатие, а указания, которые прозвучали только в разговоре, могут потеряться:

  ru code.claude.com/docs/ru/glossary

  en code.claude.com/docs/en/glossary

Если захотите обсудить

Напишите мне в Telegram: @evstratovdanil. Начните сообщение с одного слова — так я сразу пойму задачу.

«Консультация» — если нужен ответ на вопрос о курсе, команде или работе с AI: с чего начать, как поставить задачу агенту, что закрепить. Отвечаю на ваш вопрос за одну встречу до 90 минут. Условия и цена — на сайте: danilevstratov.ru/services/consulting.

«AI-агенты» — если нужна система AI-агентов для разработки курсов. Настрою её под правила вашего курса: агенты готовят черновики уроков и переводы, скрипты их проверяют, а решения принимает ваш методист. Условия и цена — на сайте: danilevstratov.ru/services/ai-agents.

Сначала созваниваемся бесплатно: уточняем задачу и выбираем формат. Это ещё не консультация. Точную цену называю письменно до старта.

Или напишите на почту: imevstratov@gmail.com

Сайт с кейсами и услугами: danilevstratov.ru