

6 принципов Claude Code: как его создатель строит рабочий сетап за 100 строк

Создатель Claude Code обходится без сложных вики-страниц и тяжёлых документов — порой буквально десятком строк в CLAUDE.md, которые он иногда удаляет целиком.

- ❏ **Важно:** Минимальный CLAUDE.md — не лень, а стратегия: чем больше инструкций, тем выше шанс, что модель запутается и проигнорирует действительно важное. Как только файл раздулся до ~100 строк — его удаляют целиком и начинают с нуля.

1. Plan Mode: сначала медленно — потом быстро

Статистика

80% сессий стартует в Plan Mode, а не сразу с написания кода

Логика

Правильный план позволяет почти не следить за агентом: он сам дойдёт до финала и подтвердит, что всё сделано

Когда необязателен

Если план уже подготовлен в другом инструменте (Cursor, Codex) с полным доступом к кодовой базе — переходить в Plan Mode необязательно

Риск без плана

Без Plan Mode агент может написать стили или архитектурные решения, несовместимые с уже существующим кодом в проекте

Правило

В рамках одной задачи стоит создавать несколько вариантов плана — это сам по себе ключевой шаг, а не формальность

2. Минимальный CLAUDE.md: меньше строк — меньше путаницы

Лимит

Целевой объём — до 100 строк; каждое обращение к файлу стоит примерно 200 тысяч токенов

Правило сброса

Когда CLAUDE.md достигает лимита — удалить целиком, затем снова добавлять инструкции постепенно по мере необходимости

Почему работает

Модели улучшаются каждый месяц: то, что нужно было прописывать полгода назад, сегодня уже зашито в модель по умолчанию

Зачем

Чем длиннее файл, тем выше вероятность, что Claude проигнорирует именно важное правило, утонувшее в объёме

3. Петля обратной связи: дай модели проверить саму себя

Принцип

Дай модели инструмент, который позволяет видеть результат её работы, и упомяни этот инструмент в промте

Примеры

Браузер через Playwright, запуск тестов, линтер — всё, что даёт объективную обратную связь без участия человека

Итерации

Несколько циклов «найди, что можно улучшить» работают значительно лучше одного прохода

Результат

Качество финального кода вырастает в 2–3 раза практически без дополнительных усилий со стороны разработчика

Дай Claude способ проверить свою работу — и он сделает это

4. Параллельные сессии: не жди — работай

Формат	Пример разбивки	Зачем	Гибрид	Риск
4–5 диалогов одновременно; каждый диалог направлен строго на одну конкретную задачу	Параллельно: создать UI-кит, создать страницу входа, обновить документацию, подготовить деплой	Агент работает автономно 30–40 минут — этот простой превращается в продуктивное время на соседних ветках	Часть диалогов — в агентном режиме с кодом, часть — в планировщике (Cursor, Codex) для проработки следующих функций	Сложно для новичков: нужно удерживать контекст нескольких задач; начинать лучше с 2 параллельных диалогов

5. Inner Loop: slash-команды для повторяющихся задач

Определение Inner loop — любая задача, которая повторяется много раз за один рабочий день, даже если кажется пустяковой	Где хранить Slash-команды лежат в <code>.claude/commands/</code> , зафиксированы в Git — доступны всей команде и не теряются	Пример: /youtube Одна команда запускает цепочку: исследование темы → подбор заголовков по аналитике прошлых видео → идеи обложек → сценарий → ТЗ на презентацию
Пример: форматирование Skill для рассылки: загружаешь сырой текст → на выходе сразу Markdown для email-рассылки, Telegram и промт для генерации картинки	Правило Любая задача, которую делаешь вручную больше двух-трёх раз в день, — кандидат на оформление в slash-команду	

Slash-команда для каждого inner loop workflow — это избавляет от повторных промтов

6. Строй под будущее: не ставь на ограничения модели

Принцип

Любые костыли и микроподкрутки ради чуть лучшего ответа через полгода, скорее всего, окажутся ненужными

Доказательство

То, на что уходили месяцы работы с n8n-агентами зимой 2024–2025, сегодня решается несколькими промтами в Claude Code

Итог

Не строй сложных систем ради самих систем – следующая версия модели, скорее всего, сделает это встроено

Правило

Never bet against the model: инвестируй время в задачи, которые не устареют с выходом следующей версии

Работайте с умом, а не с костылями

Проверь себя

☐

Начинаю ли я сложные задачи с Plan Mode, если нет готового плана с полным доступом к коду?

☐

Не превышает ли мой CLAUDE.md 100 строк – и удаляю ли я его целиком, когда он раздувается?

☐

Есть ли у модели инструмент для проверки собственного результата: тесты, браузер, линтер?

☐

Работаю ли я в 4–5 параллельных диалогах вместо последовательного ожидания одного агента?

☐

Оформлены ли мои регулярные задачи как slash-команды в .claude/commands/, зафиксированные в Git?

☐

Не трачу ли я время на сложные обходные решения под текущие ограничения модели?