



Denoise AI Lab
for education

ГАЙД

Петля: ИИ доводит работу до идеала сам

Как заставить модель саму себя проверять и переписывать — один раз объясняете, дальше система правит себя круг за кругом, пока не упрётся в потолок качества

Instagram @ai.denoise • Telegram @denoise_ai_lab

Скорее всего, вы работаете с ИИ так: дали промпт → получили результат → написали «переделай вот это» → снова получили → снова правите. И так по кругу, руками. Каждую правку держите на себе вы: вы замечаете, что не так, вы формулируете, вы гоняете модель ещё раз. Один промпт — это всегда выстрел наугад: попросили один раз и забрали то, что прилетело.

Петля работает иначе. Вы один раз объясняете, что нужно. Дальше модель сама делает работу, сама оценивает результат по чёткому стандарту, сама находит слабые места и сама переписывает — и повторяет круг, пока работа не станет по-настоящему хорошей. Вы больше не нянчите промпт, а запускаете систему, которая правит себя сама.

Почему это работает: оценить работу и переписать работу — два разных действия. Когда вы просите «сделай лучше», модель редактирует вслепую, тыкается наугад. А когда она сначала критикует собственный результат по критериям и только потом переписывает — она перестаёт гадать и целенаправленно ищет, что улучшить. В этом вся сила: критика идёт перед переписыванием, и переписывание становится осмысленным.

● Запустил и ушёл

Современные модели умеют крутить такую петлю долго и сами. Вы запускаете — и можете заняться своими делами. Модель прогоняет круг за кругом, пока не упрётся в потолок качества, и только тогда отдаёт финал.

■ Два способа собрать петлю — выберите свой

- Трек 1 — Claude Code. Если вы кодите, собираете приложение или сайт. Ставим три навыка, которые работают связкой: один пишет план, второй строит строго по плану, третий проверяет и возвращает правки.
- Трек 2 — Cework и браузер. Если работаете в обычном чате: тексты, посты, сценарии, планы, стратегии. Ставить ничего не нужно — вставляете один промпт в начало диалога, и Claude сам оценивает и исправляет себя.

■ Трек 1. Claude Code — три навыка, которые правят сами себя

Идея простая: создаём три навыка — `/spec`, `/build`, `/review` — и заставляем два последних крутиться в петле, пока сборка не пройдёт проверку начисто. Скопируйте три промпта ниже в Claude Code по очереди — каждый создаёт один навык.

1) `/spec` — превращает идею в план. Не бросается строить: сначала расспрашивает вас по одному вопросу за раз, пока не поймёт цель, требования, ограничения и то, как выглядит «готово». Потом пишет детальную спецификацию в файл.

● ПРОМПТ — НАВЫК `/SPEC`

Создай навык для Claude Code под названием «spec». Когда я запускаю `/spec`, расспроси меня про фичу или приложение, которое я хочу собрать. Задавай по одному чёткому вопросу за раз, пока полностью не поймёшь цель, обязательные требования, ограничения и то, как выглядит «готово». Не начинай строить. Когда информации достаточно — напиши понятную, детальную спецификацию и сохрани её в `specs/<name>.md`. Она должна включать: цель, точные требования, крайние случаи и конкретное определение «готово».

2) `/build` — строит строго по плану. Читает спецификацию и собирает ровно то, что в ней написано: без лишних фич, без отсебятины, без расползания задачи. В конце перечисляет, какие требования закрыл.

● ПРОМПТ — НАВЫК `/BUILD`

Создай навык для Claude Code под названием «build». Когда я запускаю `/build`, прочитай спецификацию в `specs/<name>.md` и собери ровно то, что в ней описано. Не добавляй фичи, не переписывай посторонний код и не выдумывай требования, которых нет в спецификации. Когда закончишь — перечисли, какие требования из спецификации ты закрыл, чтобы шаг проверки мог их сверить.

3) `/review` — оценивает сборку и возвращает провалы. Сверяет результат со спецификацией пункт за пунктом, выписывает каждый пробел и баг, называет требование, которому это не соответствует, и отдаёт правки обратно в `/build`. Пропускает дальше только когда выполнено всё.

● ПРОМПТ – НАВЫК /REVIEW

Создай навык для Claude Code под названием «review». Когда я запускаю /review, сравни текущую сборку со спецификацией specs/<name>.md. Иди по требованиям одно за другим и выпиши каждый пробел, баг или недоделку, называя точный пункт спецификации, которому это не соответствует. Если что-то не проходит – напиши конкретные правки и верни их, чтобы /build мог их закрыть. Пропускай сборку дальше только когда выполнены все требования спецификации.

■ Как запустить петлю (Трек 1)

- 1 Запустите /spec один раз и ответьте на его вопросы — он превратит идею в детальный план.
- 2 Вставьте промт-цикл ниже, чтобы запустить петлю /build ■ /review.
- 3 Поставьте модель на высокое усилие (high effort), запустите — и идите заниматься своими делами. Claude строит, сам проверяет, исправляет провалы и повторяет круг, пока проверка не пройдёт начисто.

● ПРОМПТ – ЗАПУСК ПЕТЛИ

Зацикль /build и /review: собирай по спецификации, проверяй сборку по спецификации, исправляй всё, что не проходит, и повторяй, пока проверка нейдёт чисто. Продолжай сам, пока нейдёт.

■ Трек 2. Cowork и браузер — один промт-петля

Ставить ничего не нужно. Вставляете один промт в начало диалога — и дальше Claude после каждой задачи сам себя оценивает и переписывает, пока не упрётся в потолок качества. Работает на текстах, постах, сценариях, офферах, планах, стратегиях — на чём угодно, где есть планка качества.

Главный трюк внутри промта — модель сначала сама пишет рубрику (по каким критериям оценивать именно эту задачу) и только потом по ней себя судит. Без рубрики оценка размытая, и петля крутится впустую. С рубрикой каждая правка бьёт по конкретной слабости.

● ПРОМПТ – ПЕТЛЯ В ЧАТЕ

С этого момента работай по петле. После каждой задачи, которую я даю, делай так: сначала составь чёткую рубрику — по каким именно критериям оценивать результат для этой конкретной задачи. Затем оцени свою работу по этой рубрике от 0 до 100. Выпиши самые слабые места и объясни, почему они стоят баллов. Перепиши работу так, чтобы закрыть главные слабости, сохранив то, что уже сильно. Повторяй петлю, пока оценка не перестанет заметно расти. Потом покажи финальную версию и как менялась оценка от прогона к прогону.

Как использовать: вставьте пром프트 первым сообщением в новом диалоге, дальше просто давайте задачу как обычно — «напиши пост про...», «сделай сценарий рилс про...», «собери план запуска...» — и Claude сам прогонит её через петлю и отдаст вычищенный финал. Можно наоборот: вставить пром프트 сразу после конкретной задачи, которую он уже сделал, — тогда петля докрутит этот один результат.

● **Лайфхак для Cowork**

Не вставляйте пром프트 каждый раз руками — сохраните его как постоянную инструкцию. Тогда петля будет включаться сама в каждом диалоге, и Claude всегда будет проверять и переписывать себя по умолчанию.

Суть одна: перестаньте нянчить промпты и соберите систему, которая делает это за вас. Больше связок и разборов работы с ИИ — в нашем каталоге и на канале Denoise AI Lab.