

Оглавление

Что я хотел бы знать до того, как чуть не бросил OpenClaw	2
1. Не гоняйте всё через лучшую модель	2
2. Агенту нужны правила. Много правил.	2
3. «Поработай над этим ночью» — так не работает	3
4. Начните с одного рабочего процесса от начала до конца	3
5. Сохраняйте то, что работает	3
6. Модель решает больше, чем всё остальное	4
7. Вы не бездарны. Это действительно сложно прямо сейчас	4

Что я хотел бы знать до того, как чуть не бросил OpenClaw

Практические советы после двух недель проб и ошибок.

1. Не гоняйте всё через лучшую модель

Это главная ошибка. Проверки состояния, cron-задачи и рутинные операции не требуют Opus или Sonnet. Настройте многоуровневую конфигурацию моделей:

- **Основная модель** — дешёвая (Haiku, Gemini Flash или локальная через Ollama) для повседневных задач.
- **Запасная модель** — мощная, подключается по необходимости.

Некоторые пользователи снизили стоимость запроса с 20–40 тыс. токенов до 1,5 тыс. просто за счёт умной маршрутизации. Переключить модель прямо в сессии можно командой `/model`.

2. Агенту нужны правила. Много правил.

Без настройки OpenClaw работает плохо: зацикливается, повторяет ответы, теряет контекст, принимает странные решения. Нужны ограничители:

- Создайте скиллы (файлы `SKILL.md` в папке `workspace/skills/`), которые явно описывают поведение.
- Добавьте правила против зацикливания, инструкции по сжатию контекста, проверку задач перед вопросами к пользователю.
- Агенты, которые работают хорошо — это агенты с детально настроенными инструкциями.

Вы должны сами исследовать и понимать, что делаете. Не надейтесь, что агент знает всё. Вы — дирижёр, так дирижируйте.

3. «Поработай над этим ночью» — так не работает

Если дать задачу и закрыть чат — агент всё забудет. Сессии сохраняют состояние только пока открыты.

Для фоновой работы нужны:

- **Cron-задачи** с изолированными сессиями — запускают независимые экземпляры агента по расписанию и отправляют результаты.
 - **Очередь задач** (Notion, SQLite, текстовый файл) в паре с cron, который проверяет очередь.
-

4. Начните с одного рабочего процесса от начала до конца

Не пытайтесь настроить почту + календарь + Telegram + веб-скрапинг + cron одновременно. Каждая интеграция — отдельная точка отказа.

- Добейтесь идеальной работы **одного** воркфлоу (например, утренний брифинг по cron).
 - Только потом добавляйте следующий.
 - При проблемах запустите `openclaw doctor --fix`.
-

5. Сохраняйте то, что работает

Сжатие контекста (compaction) со временем теряет информацию. Используйте:

- **Файлы состояния** — фиксируйте прогресс.
- **Документы рабочего пространства** — заполняйте `USER.md`, `AGENTS.md`, `HEARTBEAT.md`.
- **Постоянное хранилище** — записывайте важные решения.

Чем меньше агенту приходится переучиваться, тем лучше он работает.

6. Модель решает больше, чем всё остальное

Большинство проблем — из-за моделей, которые плохо работают с вызовами инструментов. Качество в чате и качество в агентном режиме — это разные вещи.

Модель	Вердикт
Claude Sonnet / Opus	Надёжно работают с tool calls
GPT-5.2	Хорошо справляется
Kimi K2 (через API)	Хорошо справляется
DeepSeek Reasoner	Отличное рассуждение, но формирует некорректные вызовы инструментов
GPT-5.1 Mini	Очень дёшевая, но практически бесполезна для агентной работы

7. Вы не бездарны. Это действительно сложно прямо сейчас

OpenClaw — не готовый продукт. Люди, которые пишут «мой агент собрал приложение за ночь», потратили недели на настройку. Разрыв между демо и реальным использованием существует. Он сокращается, но пока ещё ощутим.

Надеюсь, это поможет кому-то до того, как он решит всё бросить.