🚀 Harness: почему обвязка важнее самой AI-модели

🎯 Harness: почему в 2026 году обвязка важнее самой AI-модели

Ключевые тезисы:

- **Harness (упряжка)** — это набор правил, инструментов, памяти, проверок и циклов, который оборачивает AI-модель, превращая её из «текстового процессора» в автономного агента, способного часами работать самостоятельно.
- Одна и та же модель в разных харнесах может показывать разницу в результативности в **десятки раз** (например, с 6,7% до 68% решённых задач).
- Промтинг и контекст-инжиниринг уступают место **инженерии харнесов (Harness Engineering)** как новой парадигме работы с AI.
- В 2027 году ожидается переход к **динамическим харнесам**, которые агенты будут создавать под конкретную задачу на лету.

🐎 Что такое Harness и почему он важен?

Harness (англ. «упряжь») — это метафора для всей инфраструктуры, которая направляет и усиливает AI-модель. **Модель (Claude, GPT, Gemini) — это мощная, но безвольная «лошадь».** Сама по себе она лишь генерирует текст.

Harness — это «упряжка», которая включает:

- **Цикл агента** (агент принимает решение → харнес выполняет → возвращает результат).
- **Инструменты (Tools)** — доступ к терминалу, файлам, Git, веб-поиску.
- **Субагенты** — параллельные помощники с отдельными контекстами.
- **Правила, проверки, долгую память, интеграции (MCP)**.

- **Песочницу и разрешения.**

Без харнеса модель бесполезна для сложных задач. С ним — это автономный работник.

💡 **Пример:** В Claude AI и Claude Code — одна и та же модель, но разный харнес. В чате она пишет текст, а в Code — правит код, запускает тесты и читает логи.

📈 Эволюция подходов к работе с AI

1. 🔥 **Эпоха промт-инжиниринга (2023-2024):** Искусство составления «заклинаний»-промтов для моделей с маленьким контекстом (~4K токенов).
2. 🔄 **Эпоха контекст-инжиниринга (2024-2025):** Рост контекстного окна (до 1M токенов). Появление RAG, MCP, вызовов инструментов. **Проблема:** «гниение контекста» (context rot) — чем больше заполнено окно, тем «глупее» становится модель, она начинает «врать» и «забывать» задачи.
3. 🚀 **Эпоха инженерии харнесов (Harness Engineering, с 2025):** Фокус сместился на создание правильной «обвязки» вокруг модели. На каждом шаге — свежий контекст, жёсткие правила, внешние проверки, инструменты. Это лечит проблемы предыдущих эпох.

📊 Цифры, доказывающие силу Harness

1. **Эксперимент (февраль 2026):** Одна модель (Grok Coder Fast) с разными форматами харнеса показала результат **6,7%** против **68%** решённых задач.
2. **Исследование Stanford (март 2026):** Смена харнеса (без смены модели) дала **+7,7 п.п.** качества, **в 4 раза меньше токенов** и **+4,7 п.п.** точности на сложных задачах.
3. **Промт vs. Harness:** В реальных проектах улучшение промта даёт **<3%** прироста, а изменение харнеса — **десятки процентов**.
4. **Пропорции в Claude Code (утекшие исходники):** **60%** кода — логика модели, **40%** — логика харнеса.

5. **Компромисс Antropic:** Соло-агент: 20 мин / $9. Полный харнес (3 агента): 6 часов / $200. **Дороже в 20 раз, но качество лучше на порядок.**

🎯 **Вывод аналитика Акаша Гупты (январь 2026):** *«Модель — это товар (commodity), как пшеница или нефть. А Harness — это ров вокруг крепости (moat), ваше конкурентное преимущество».*

🛠️ Топ-7 готовых Harness на рынке (2026)

1. Claude Code & Claude Agent SDK (Antropic)

- **Где:** Терминал, десктоп-приложение, VS Code, веб-версия.
- **Суть:** Готовый, отполированный харнес «из коробки» вокруг моделей Antropic (и любых через API).
- **Что внутри:** Цикл агента, инструменты (bash, поиск), субагенты, хуки (автопроверки), файл `.claude.md` (долгая память), MCP-серверы, система разрешений.
- **Фишка (май 2026): Dynamic Workflows** — Claude сам пишет JavaScript-оркестрацию для задачи, запуская сотни субагентов параллельно.
- **Кому:** Разработчикам и всем, кто хочет, чтобы AI писал код. Самый отшлифованный вариант.

2. OpenAI Codex

- **Где:** Терминал, VS Code, веб, macOS, iOS. Единый **Codex App Server**.
- **Суть:** Альтернатива Claude Code от OpenAI для своей экосистемы.
- **Рост:** С 82 тыс. скачиваний (апрель 2025) до **14,5 млн** (март 2026).
- **Особенность:** Модели OpenAI обучены на **patch-формате** (стиль git diff), Antropic — на **string replace**. Нельзя переключать модели в середине задачи (падение качества).
- **Кому:** Тем, кто уже в стеке OpenAI и хочет один счёт/интерфейс.

3. Cursor (Anycode)

- **Где:** Собственная IDE (на базе VS Code) + Cursor Agent CLI.

- **Суть:** Harness, **вшитый в ядро редактора**. Не подключается к вашей среде — вы меняете среду на него.
- **Что внутри:** Индексация всего репозитория, Cloud Agents (работают в облаке), параллельные сессии, Slack/GitHub-интеграции.
- **Фишки:** Собственные модели (**Composer 2.5** для многофайловых правок, **Cursor Tab** — предсказание действий в редакторе, **Backbot** — AI-ревьюер PR).
- **Кому:** Разработчикам, которые предпочитают работать в IDE, а не в терминале. Доверяют >50% компаний из Fortune 500.

4. Devin (Cognition)

- **Где:** Полностью облачный автономный агент + десктопная IDE (Windsurf).
- **Суть:** Harness для **полного делегирования**. Даёте задачу — уходите — получаете готовый PR. Рассчитан на работу без вашего вмешательства.
- **Что внутри:** Автономный цикл, песочница под каждую задачу, встроенный браузер, планировщик, ICU (единица работы для биллинга).
- **Цена:** Дорого (сотни $ в месяц при активной работе).
- **Кому:** Менеджерам, техлидам, стартапам для делегирования больших, чётких задач (миграции, написание тестов). Не для точечного кодинга.

5. Google Антигравити & Agent Development Kit (ADK)

- **Где:** Google Cloud + Gemini-модели.
- **Суть: Antigravity** — готовый агентский IDE (аналог Cursor). **ADK** — open-source фреймворк для сборки своих агентов.
- **Что внутри (Antigravity 2.0):** Мультиагентная оркестрация (Planner, Executor, Verifier), встроенный браузер (Chromium), динамические субагенты, фоновые задачи, глубокая интеграция с Google Cloud.
- **Кому:** Крупному корпоративному бизнесу, уже сидящему на Google Cloud (банки, ритейл, телеком).

6. LangGraph

- **Где:** Ваш сервер, ваш код.
- **Суть: Конструктор** для сборки своего харнеса, а не готовый продукт. Часть экосистемы LangChain.

- **Модель:** Графовая (ноды — шаги, рёбра — условия). Python-based.
- **Что можно собрать:** Сложные циклы, ветвления, параллельное выполнение, чекпоинты состояния, human-in-the-loop.
- **Кому:** Разработчикам, которые строят кастомных агентов под специфичные бизнес-задачи (боты, интеграции с CRM), когда готовые решения не подходят.

7. CrewAI / Autogen / AI2

- **Где:** Ваш сервер.
- **Суть: Мультиагентные фреймворки**, где агенты с ролями (PM, инженер, дизайнер) общаются между собой.
- **Проблема: Компаундинг ошибок**. Если каждый агент надёжен на 95%, то цепочка из 5 агентов будет надёжна лишь на ~77%. Точность падает с ростом цепочки.
- **Кому:** В основном исследователям и экспериментаторам. Для 90% бизнес-задач один хорошо настроенный агент с инструментами работает лучше.

🇷🇺 Бонус: Российский Harness — GigaChain (Сбер
