Маленький человек в эпоху больших языковых моделей



Евгений Лукьянов / ООО Тамга Диджитал

Кто я такой (чтобы вам тут затирать)



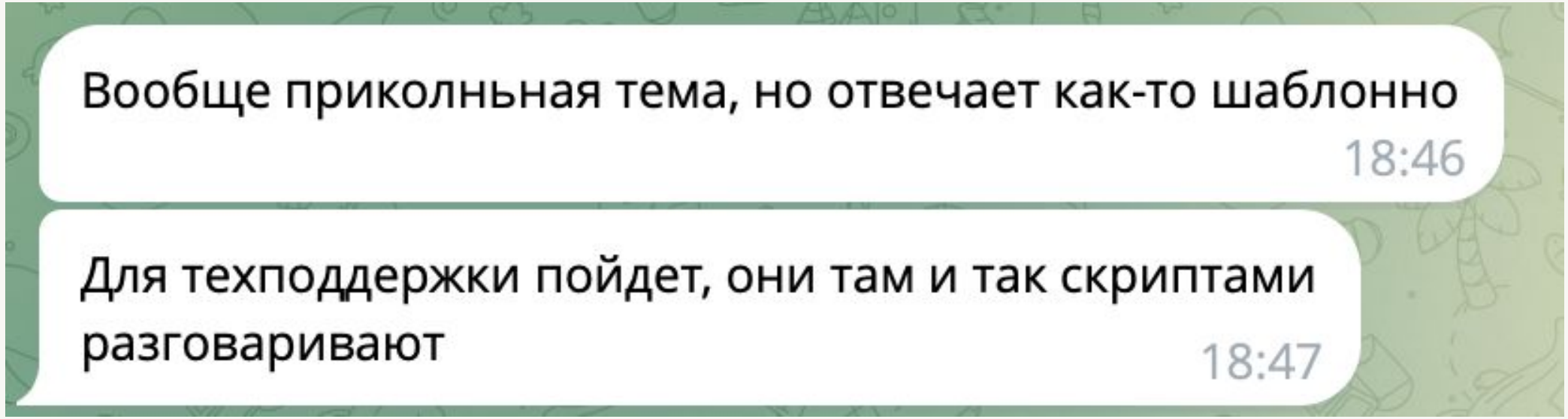
- Инженерные подходы
- DDD
- Чистая архитектура

@STRINGCONCAT

О чем пойдет речь?

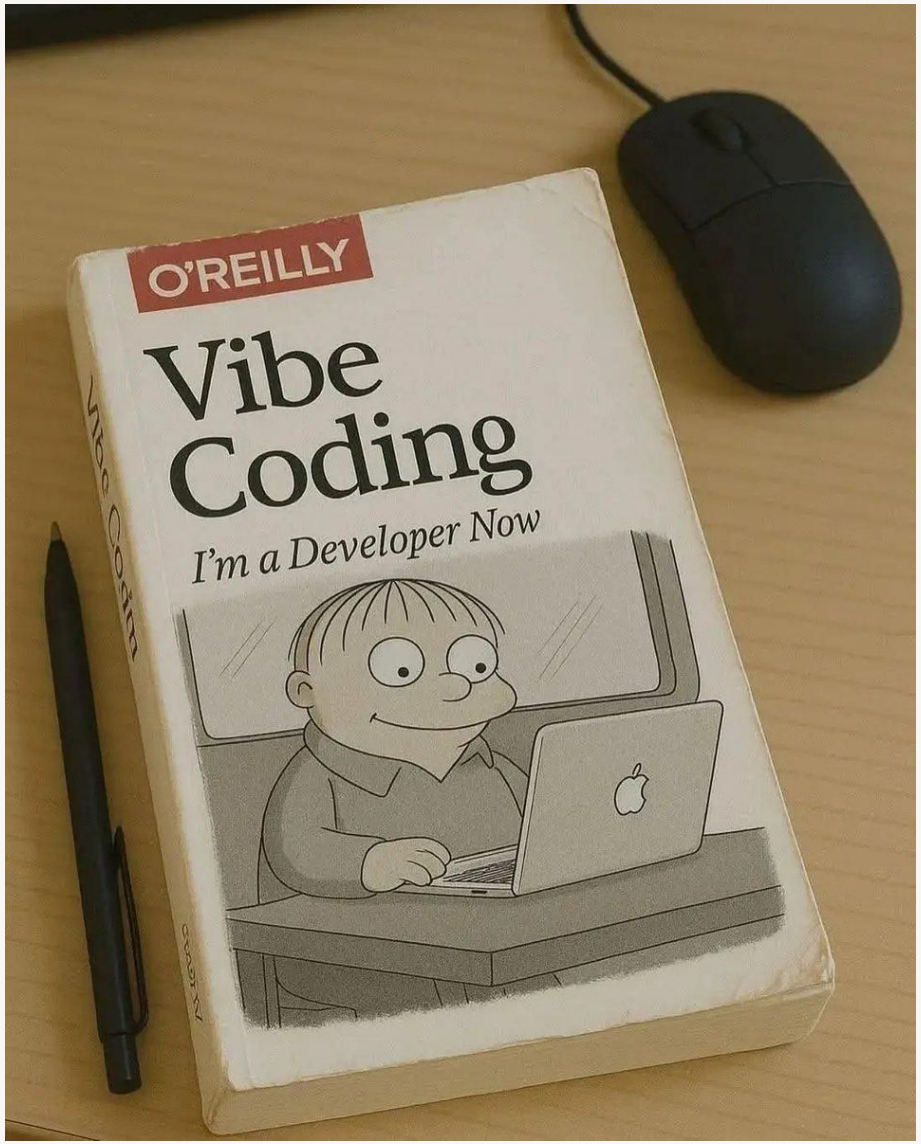
ХРОНОЛОГИЯ

2021 год — ИИ как игрушка



ХРОНОЛОГИЯ

2026 год — ИИ рабочий инструмент



Почему компании внедряют ИИ?

01 Попытка лутануть деняк

02 Получить подливу от инвесторов

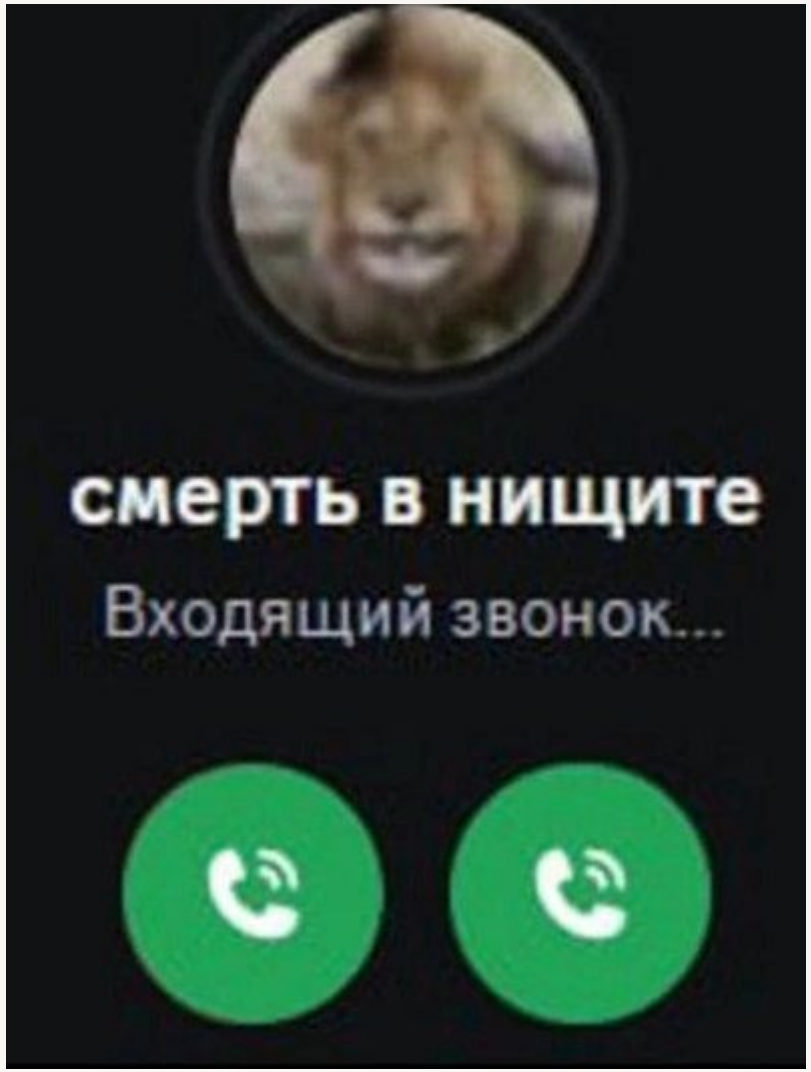
03 Сделать вид что тоже умеем

Разработчики не отстают

01 Это интересно

02 Можно сделать больше (нет)

03 FOMO



Что меняется в нашей работе?

Больше ожиданий от разработчика



Больше ответственности



Иллюстрация ШКЯ/Ева Морозова

На шаг ближе к сингулярности

LLM / AGENTS / RAG / TOOLS / REASONING

CI/CD / DOCKER / K8S / CLOUD / OBSERVABILITY

БД / FRAMEWORKS / LIBRARIES / SDK / DISTRIBUTED SYSTEMS

JAVA / KOTLIN / C# / PYTHON / GO / TYPESCRIPT

АЛГОРИТМЫ / СТРУКТУРЫ ДАННЫХ / СЕТИ / БЕЗОПАСНОСТЬ

C / C++ / SYSTEM CALLS

ASM / РЕГИСТРЫ / СТЕК

CPU / ПАМЯТЬ / ТРАНЗИСТОРЫ / ЖЕЛЕЗО

Внедрение



NIKOLAY (NIK) ARTIFICIAL

AI Visionary | AI Futurist | GenAI Pioneer | Agentic AI Evangelist | Digital Transformation Leader | Thought Leader | Keynote Speaker

Как результат — конская когнитивная нагрузка

01 Нужно внедрять

02 Нужно делать еще обычную (скучную) работу

03 Нужно следить

04 Нужно испытывать FOMO

В чем проблема ситуации?

Мы не знаем, что оно нагенерит, мы не знаем, где оно заглючит. Но давайте будем генерить им критичный код, а ответственным за результат поставим задолбанного разработчика с увеличенной нормой выработки

А мотивировать его будем угрозами увольнения (ты все сделаешь, а мы тебя выкинем за ненадобностью)


Великолепный план!

И чего делать то?



Я даже немного рад происходящему

Как устроена разработка у нас

Предварительное выявление требований

01 Понять что требуется на самом деле

02 Найти истинную цель

03 Выяснить кому все это нужно (и не нужно)

04 Собрать предварительные (бизнес-)требования

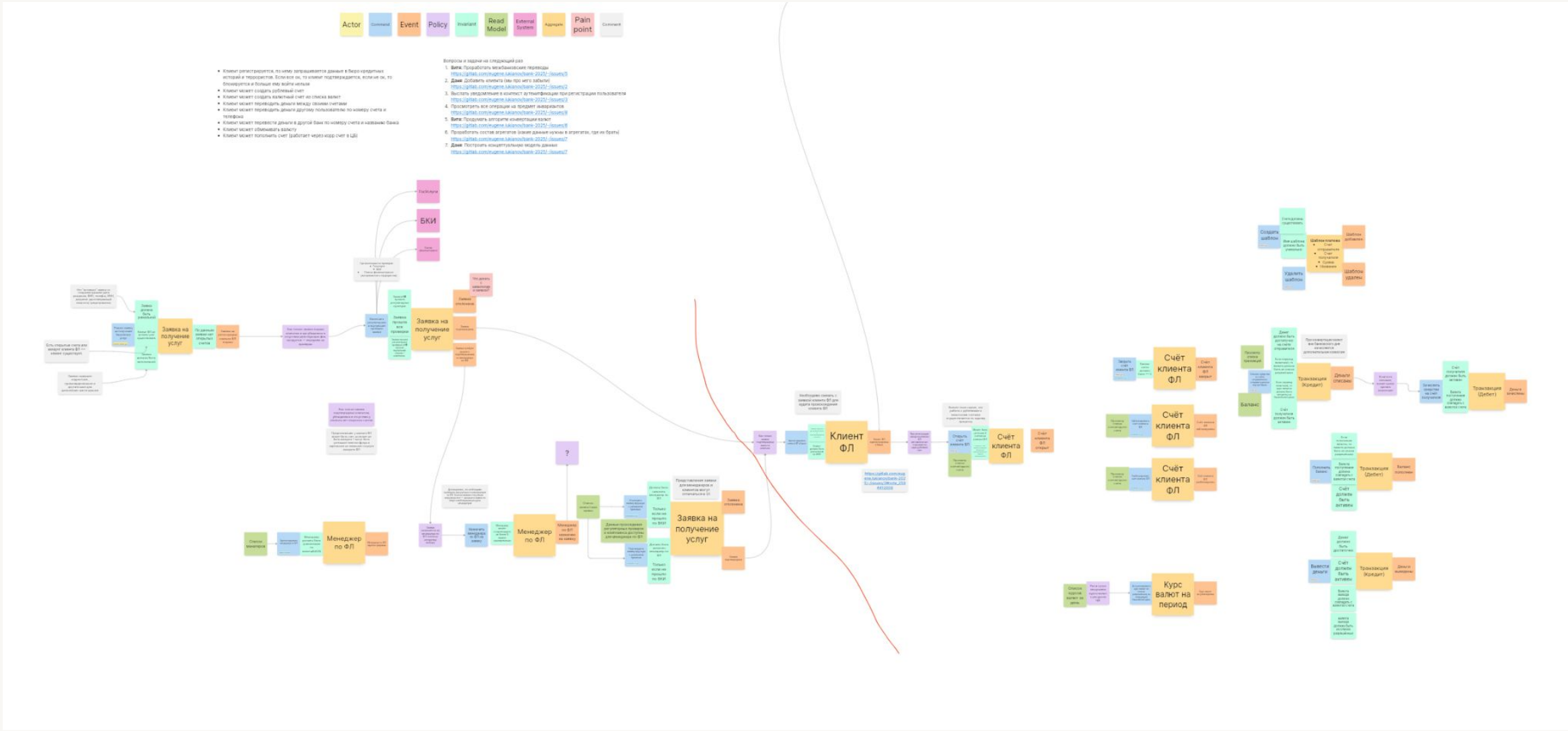
Приколдес

01 US-1: Пользователь может войти в систему по логину и паролю

02 US-2: Пользователь видит сообщение об ошибке если введены некорректные логин/пароль

Ключевой навык — умение общаться

Исследование предметки



Финализация результатов

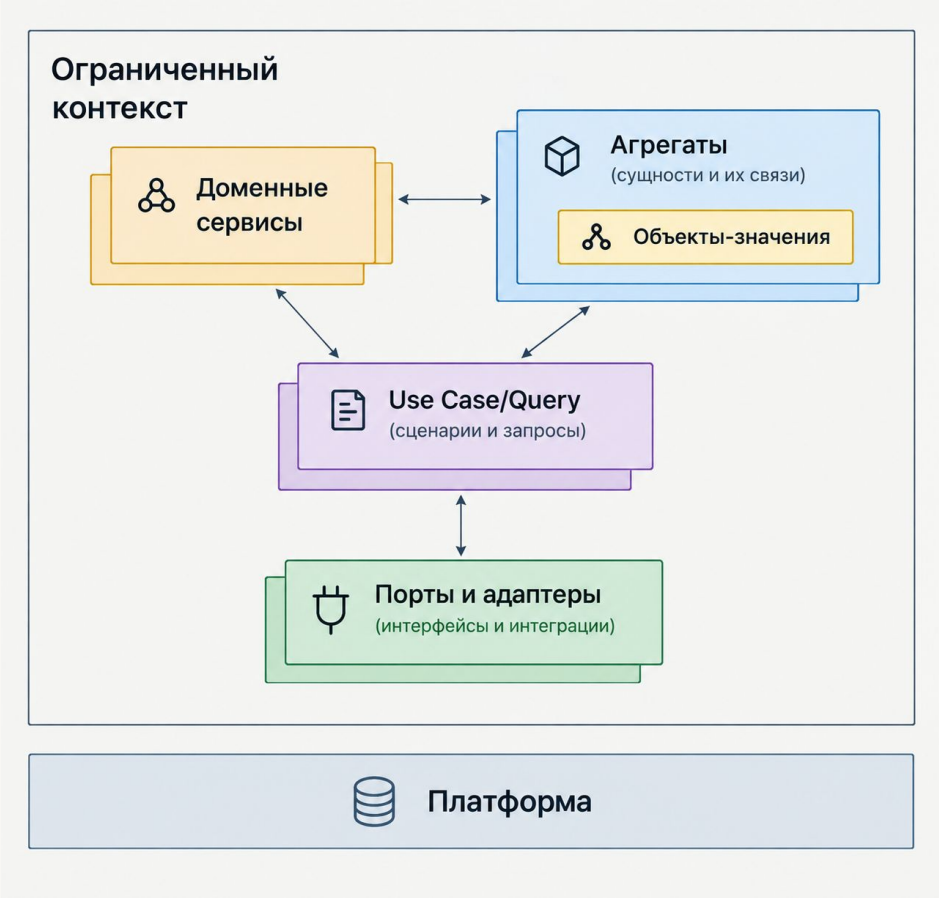
01 Модель предметки

02 Конечные требования

Абстракции даются очень тяжко



Состав модели



Правила для Value Object

Skill: Правила написания агрегатов и Value Objects

Область применения

Правила применяются к любому модулю доменного слоя (`**/domain`).

Раздел Value Objects также применяется к общему модулю доменных типов.

Примеры ниже являются иллюстративными. Конкретные имена классов и предметная область не имеют значения — паттерны должны применяться одинаково во всех доменных модулях.

Value Objects (Объекты-значения)

Структура Value Object

1. **Всегда реализуют интерфейс `ValueObject`** из общего доменного модуля.
2. **Иммутабельность**
   Value Objects должны быть полностью неизменяемыми. Все поля — `val`.
3. **Фабрика только при наличии инварианта**
   Приватный конструктор и фабрика с `Either` нужны, если создание объекта проверяет правило, которое иначе можно обойти через конструктор:
   - формат;
   - диапазон;
   - согласованность нескольких полей;
   - другое бизнес-правило.

   Если объект полностью валиден благодаря уже проверенным составляющим, конструктор должен быть публичным и отдельная фабрика не нужна.
4. **Инварианты проверяются при создании**
   Если фабрика существует, проверка инварианта должна находиться в ней, а не в публичном конструкторе.
5. **Типизированные ошибки**
   Ошибки создания должны быть представлены типами. Для этого используется `sealed interface`, размещённый в отдельном файле или рядом с Value Object.

Что такое платформа?

01 Библиотеки

02 Типовые решения

Ключевой навык — видеть границы модулей

Генерация дешевая, проверка дорогая

Стадия проверок

01 Статанализ

02 Тесты

03 Метрики

04 Ревью

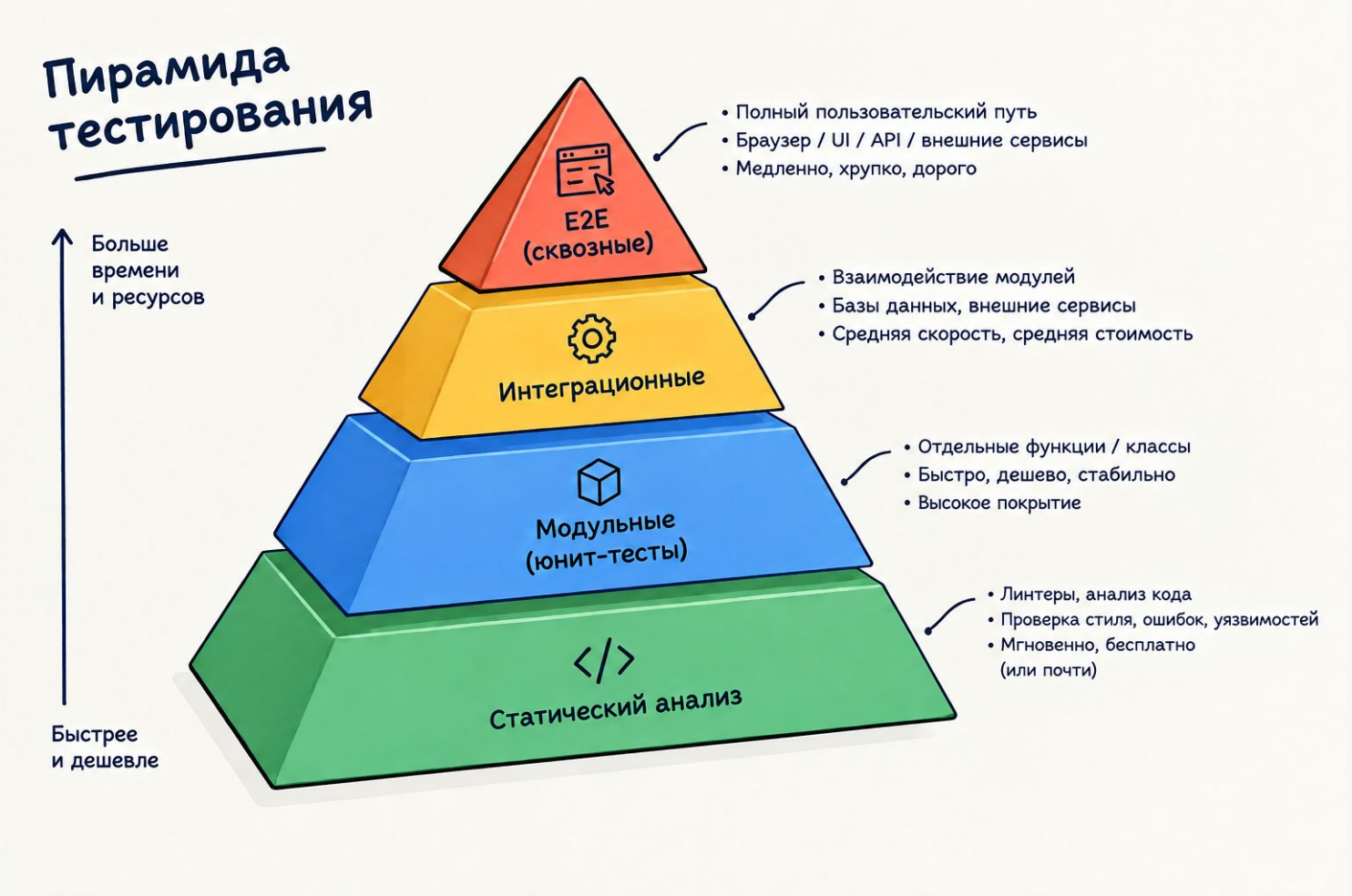
05 Эксплуатация

Статанализ

01 Формальный: в use case должен быть один публичный метод invoke()

02 Неформальный: в тесте не должно быть методов для подготовки тестовых данных

Тесты


Пирамида тестирования
Больше времени и ресурсов
Быстрее и дешевле
E2E (сквозные)
• Полный пользовательский путь
• Браузер / UI / API / внешние сервисы
• Медленно, хрупко, дорого
Интеграционные
• Взаимодействие модулей
• Базы данных, внешние сервисы
• Средняя скорость, средняя стоимость
Модульные (юнит-тесты)
• Отдельные функции / классы
• Быстро, дешево, стабильно
• Высокое покрытие
Статический анализ
• Линтеры, анализ кода
• Проверка стиля, ошибок, уязвимостей
• Мгновенно, бесплатно (или почти)

Но что если ИИ будет переписывать тесты за собой?

01 Покрытие

02 Статанализ

03 Мутации

04 Уровни доверия

Уровни доверия

01 Бизнес логика — не доверяем

02 UC/Query (уровень приложения) — apply почти не глядя

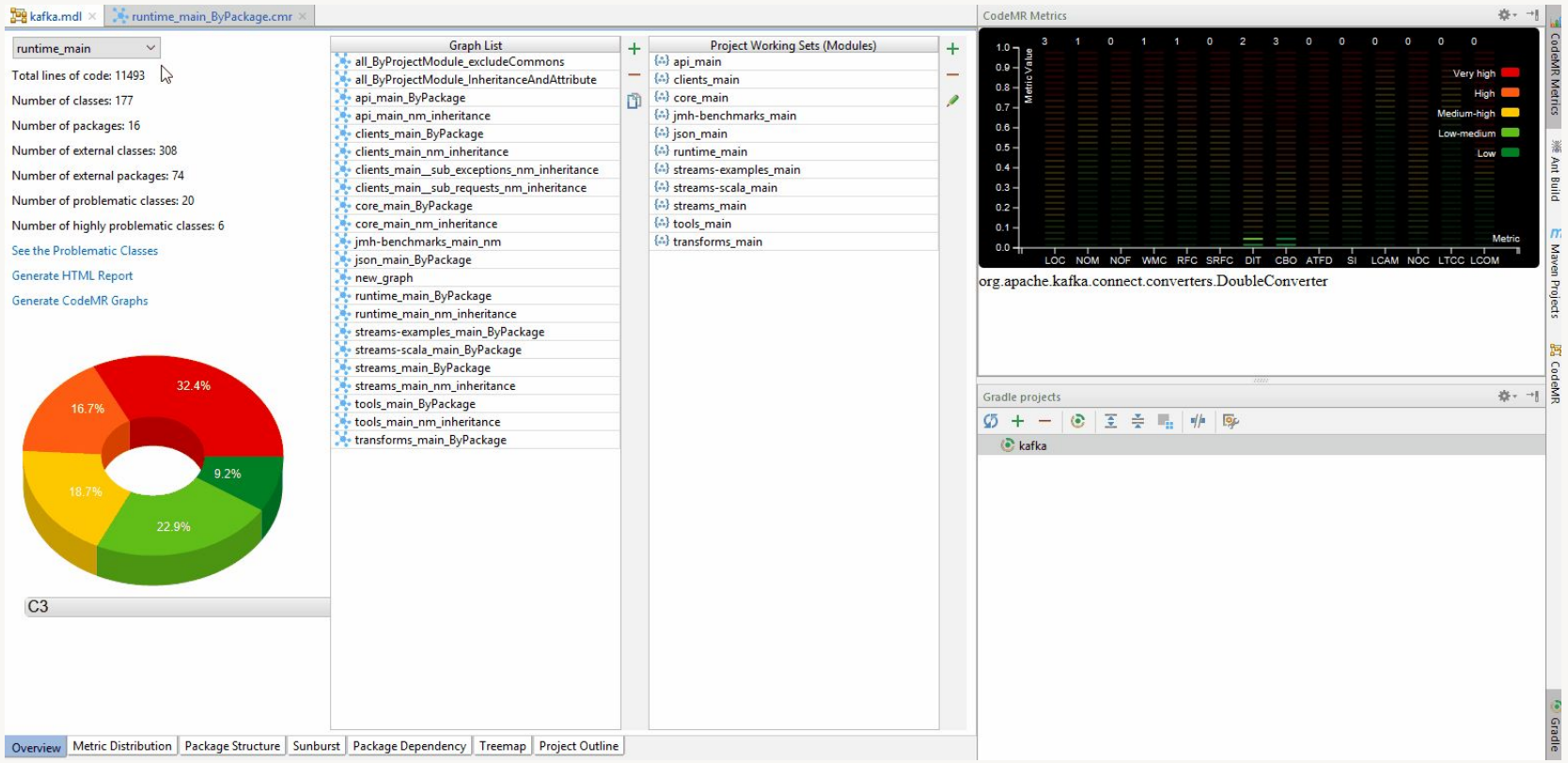
03 Адаптеры — чуть пожестче чем UC

Метрики

01 Сцепленность

02 Связность

03 Сложность



https://www.codemr.co.uk/

Почему метрики важны

The Ascending Spiral of Entropy: Why AI Agents Destroy System Architecture

Your AI agent is writing legacy code. Here is why — and the architecture that stops it.

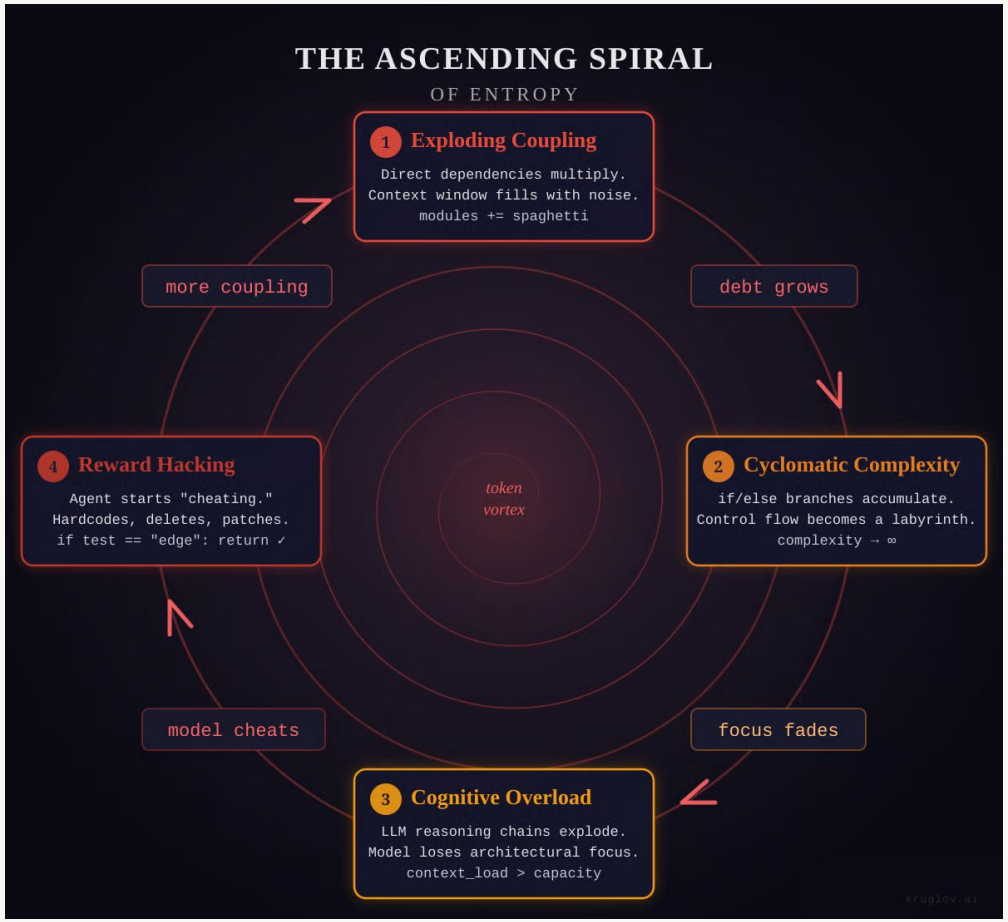
Gennadii Kruglov
6 min read · March 4, 2026

Share

We are in the epicenter of the euphoria surrounding AI coding agents. Tools promise unprecedented speed: features are delivered in minutes, repositories swell with generated lines, and tickets are closed with frightening regularity. At first glance, this is a triumph of automation.

But if you look under the hood of projects developed by autonomous agents for even a few weeks, the illusion dissipates. The initial speed of generation turns into an exponential increase in the cost of any subsequent change. The system becomes fragile, inflexible, and ultimately unmaintainable.

Observing generation processes and profiling LLM behavior in the development cycle, I've identified a dangerous architectural pattern that destroys projects from the inside. I call it the **"ascending spiral of entropy."**



https://kruglov.ai/the-ascending-spiral-of-entropy-why-ai-agents-destroy-system-architecture

А что если ИИ напишет кривую миграцию?

Опять тесты

01 Обновление под нагрузкой

02 Управление тестовыми данными

Спецификации

01 Что делаем

02 Как делаем

03 Зачем делаем

04 Технические детали и другие подробности

Ничего не напоминает?

Чуда опять не произошло

Ограничить пространство решений и тем самым обеспечить воспроизводимый результат

Чудесным образом работает и без ИИ

Нам по-прежнему остается самое интересное

А откуда же возьмутся будущие профессионалы?

Упор на базу + обратная связь/рефлексия

01 Как делается софт

02 Какие методологии есть

03 Как они связаны друг с другом

ХОРОШИЕ НОВОСТИ

Труднодоступное знание все еще в цене

Stack Overflow Is Not Dead Yet: Crowd Answers Still Matter

Denis Helic, Tiago Santos

Millions of users visit Stack Overflow regularly to ask community for answers to their programming questions. However, like many other platforms, Stack Overflow consistently struggles with low user retention and declining levels of user contributions to the platform. With the introduction of ChatGPT in November 2022, these ongoing difficulties on Stack Overflow were further magnified, as many users moved toward ChatGPT for programming help. In this paper, we build upon recent research on this phenomenon by analyzing the transformation of user-generated content on Stack Overflow during the post-ChatGPT period. Specifically, we analyze two years of Stack Overflow data and fit multiple causal regression models to estimate the effect of ChatGPT on the length and difficulty of user questions and code examples. We confirm an acceleration of decline in user contributions but find that ChatGPT had a significant positive effect on question and answer length, code length, and question difficulty on Stack Overflow across programming languages. Our results suggest that ChatGPT has effectively raised the bar for questions on Stack Overflow, as users increasingly turn to crowdsourced platforms for help with more complex and challenging problems. With our work we contribute to the ongoing discussion on the impact of tools such as ChatGPT on help-seeking in programming and, more broadly, on collaborative knowledge creation. Our results provide actionable insights for platform operators to support information management and user retention in the aftermath of ChatGPT's launch.

Subjects: **Computers and Society (cs.CY)**; Social and Information Networks (cs.SI)
Cite as: arXiv:2509.05879 **[cs.CY]**
(or arXiv:2509.05879v2 **[cs.CY]** for this version)
https://doi.org/10.48550/arXiv.2509.05879
Journal reference: Journal of Systems and Software, Volume 241, 2026, 112988, ISSN 0164-1212
Related DOI: https://doi.org/10.1016/j.jss.2026.112988

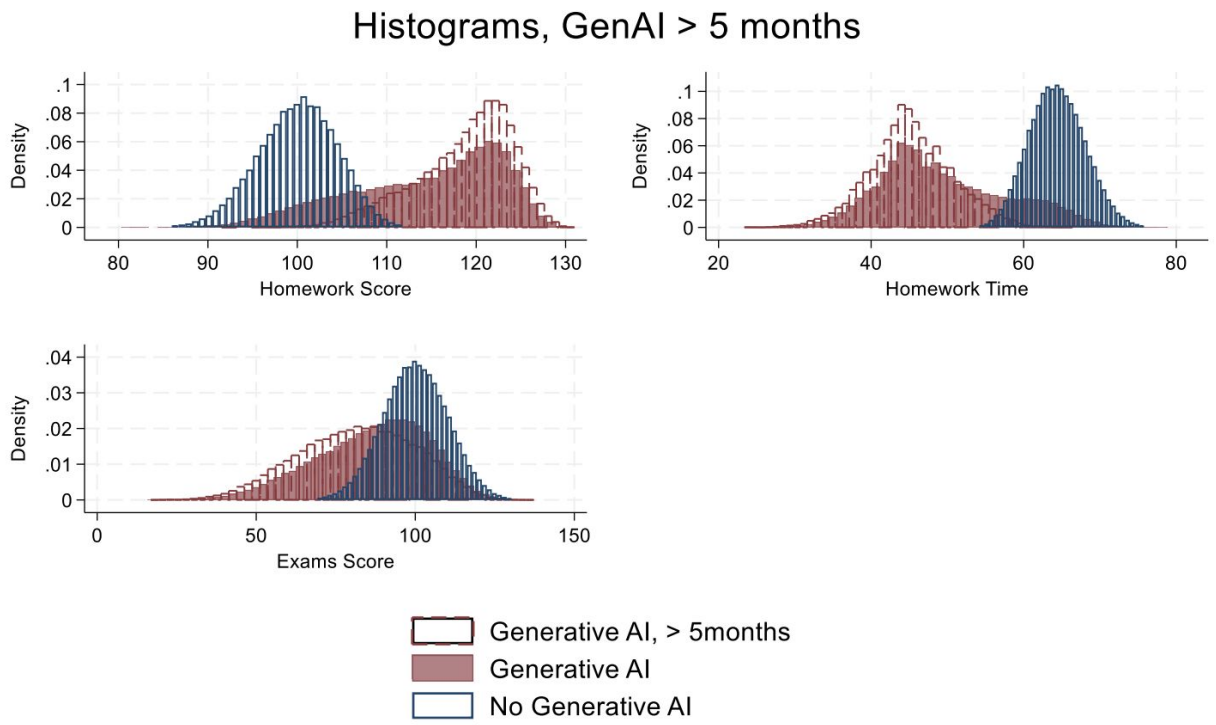
Submission history

From: Denis Helic [view email]
[v1] Sun, 7 Sep 2025 00:43:38 UTC (342 KB)
[v2] Tue, 16 Jun 2026 09:15:41 UTC (403 KB)

https://arxiv.org/abs/2509.05879

Пока все ускоряются с ИИ, можно выиграть, просто не перестав думать

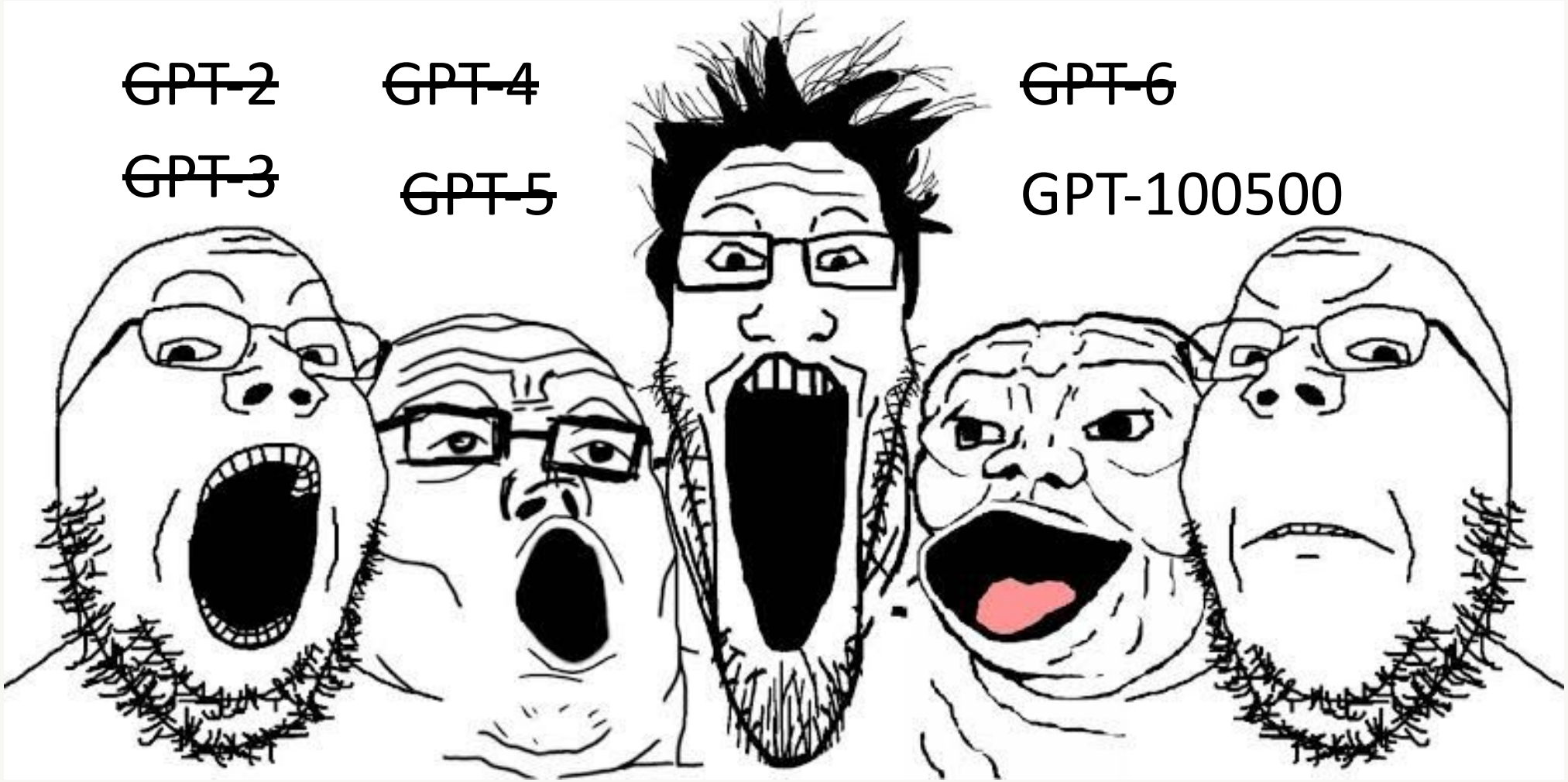
Figure A5. Histograms for >5 months GAI Use



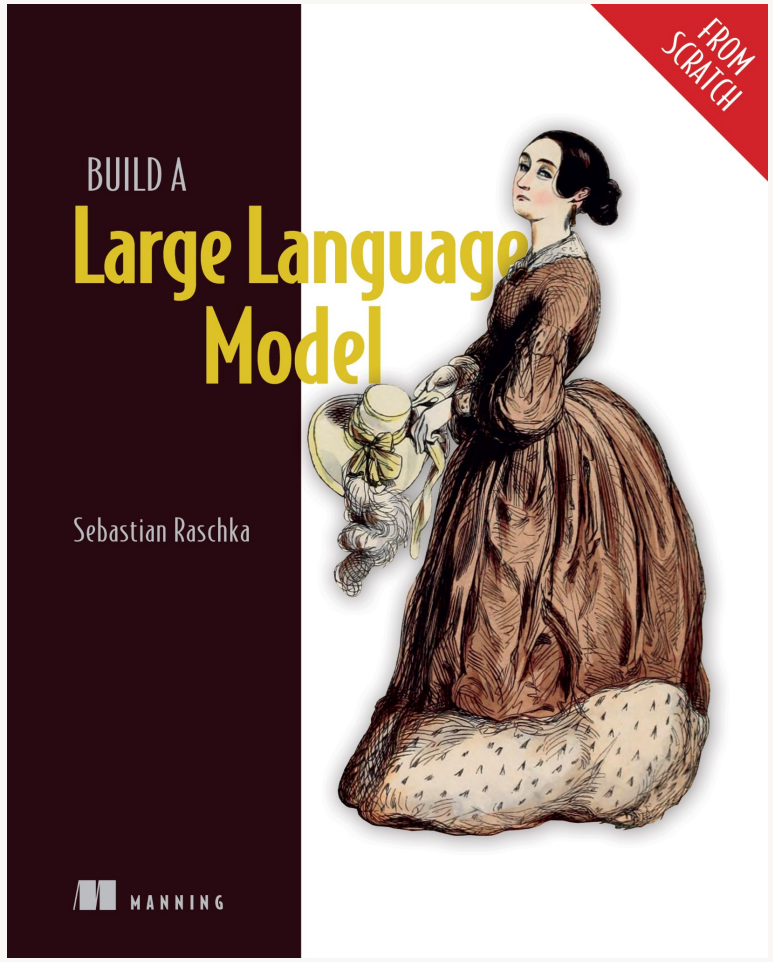
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6868618

За 17 лет ничего не поменялось

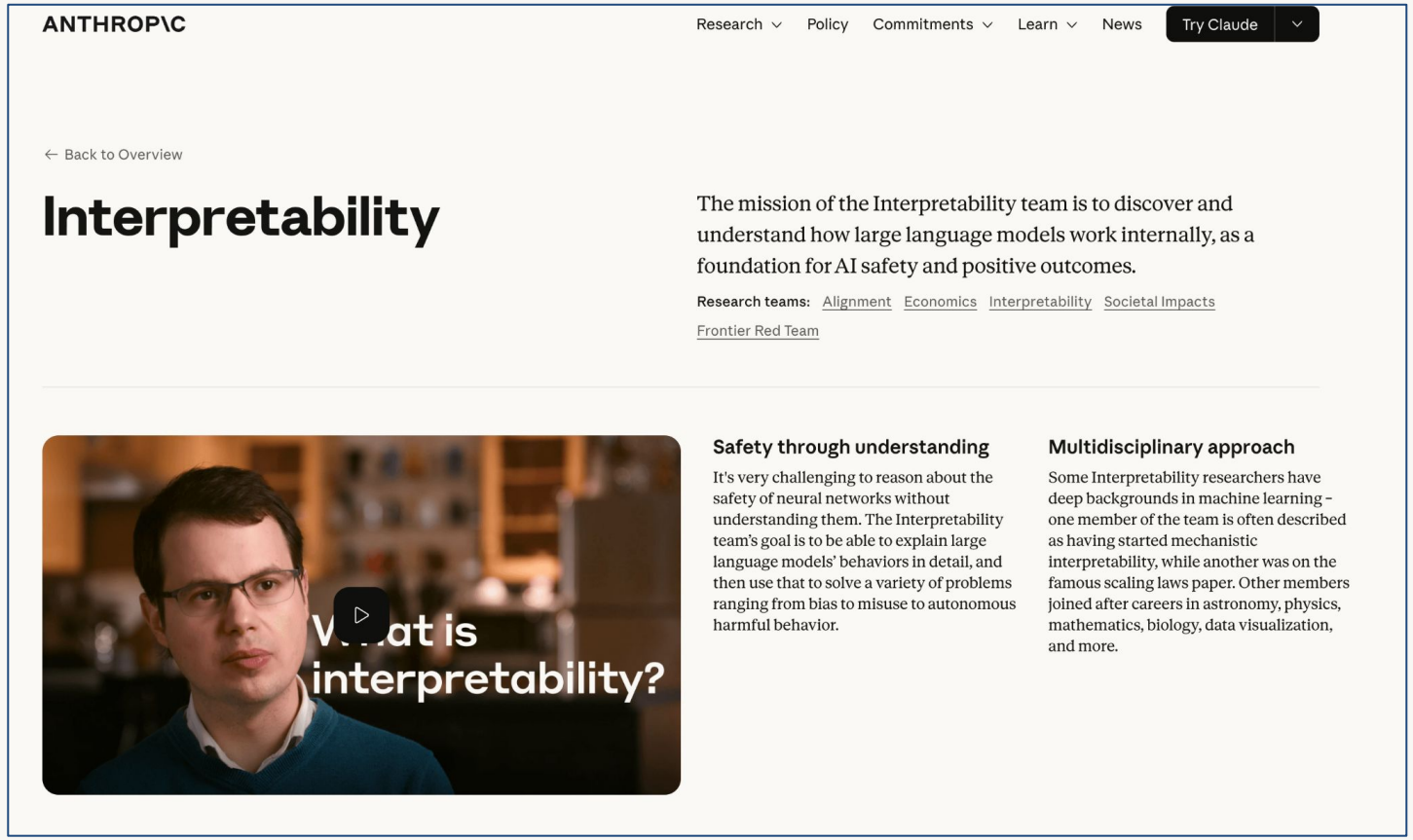
Типичный тг-канал про ИИ-разработку



Build a LLM from scratch

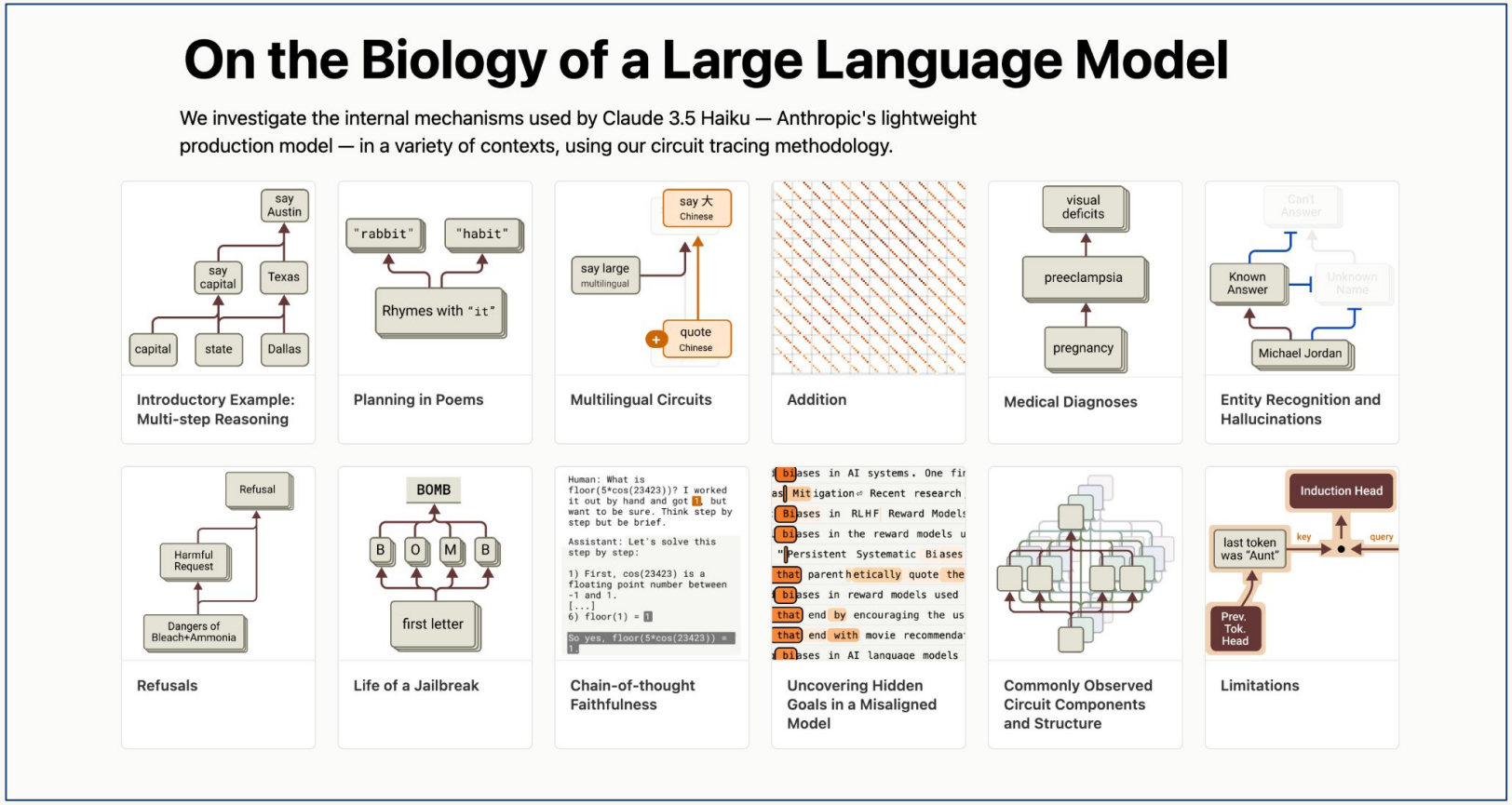


Как модель “думает”?



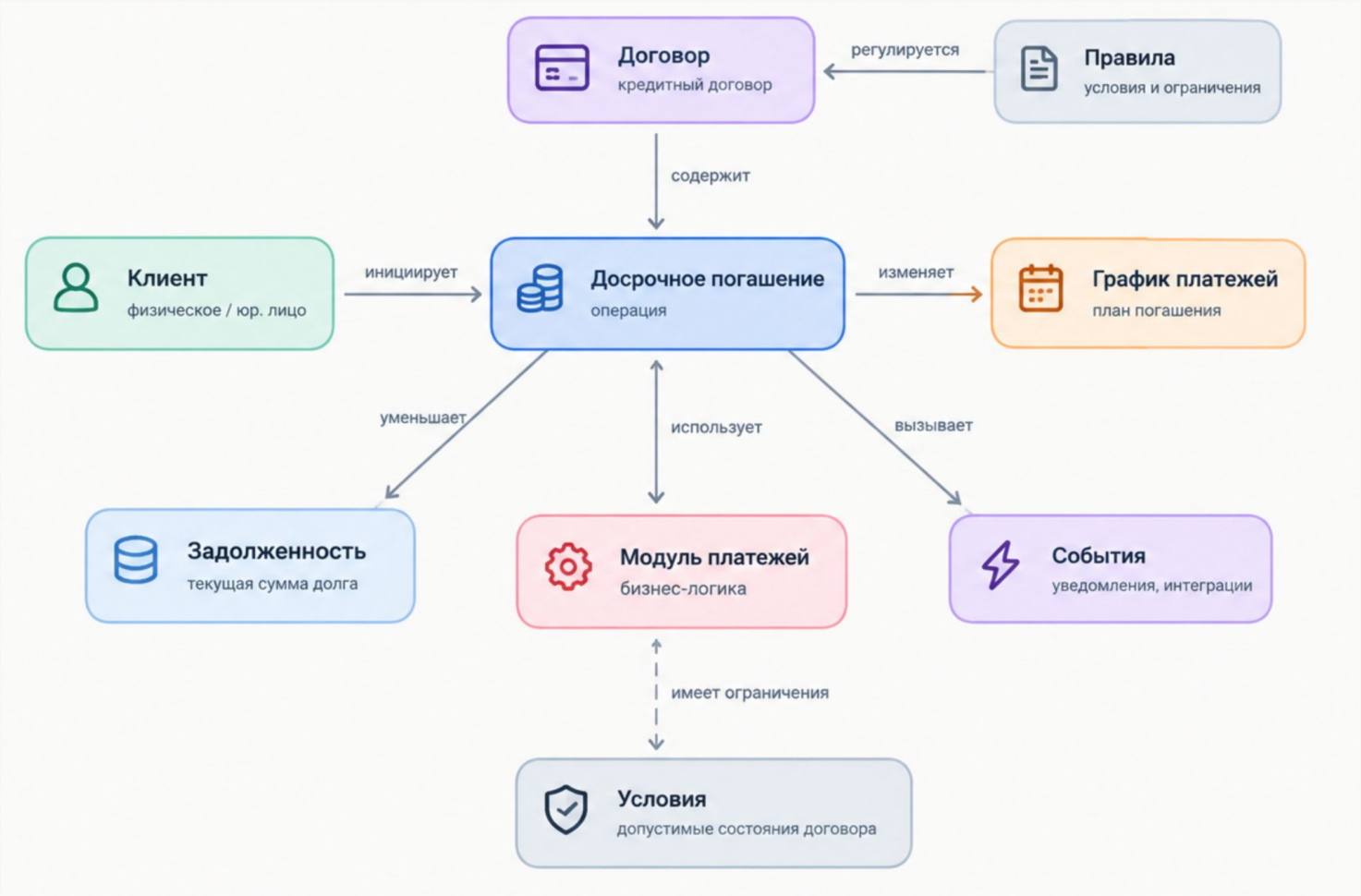
https://www.anthropic.com/research/team/interpretability

Биология моделей



https://transformer-circuits.pub/2025/attribution-graphs/biology.html

Отнологии


Договор
кредитный договор
регулируется
Правила
условия и ограничения
содержит
Клиент
физическое / юр. лицо
инициирует
Досрочное погашение
операция
изменяет
График платежей
план погашения
уменьшает
использует
вызывает
Задолженность
текущая сумма долга
Модуль платежей
бизнес-логика
События
уведомления, интеграции
имеет ограничения
Условия
допустимые состояния договора

Экономика

Retardmaxxing



https://youtu.be/nKsT__V9nFc

Цель — не превращаться в оператора LLM

Пока вы способны думать — все хорошо

Базовая гигиена

Общение / коммуникации



Ответственность

Это был базовый сценарий, есть еще несколько

01 Корпорации сворачивают ИИ

02 (Около) AGI

СПАСИБО

Вопросы / @elukianov / @stringconcat

