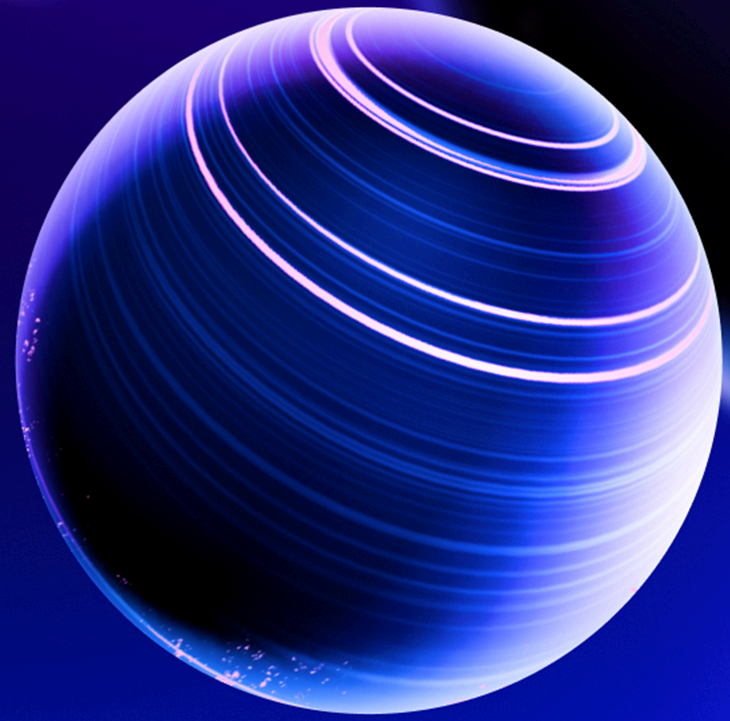
GIT VERSE

# На пути
# К 100% генерации кода



# Developer-first платформа с AI, доступная в облаке

# GITVERSE

# ПРАВИЛО БЕЗОПАСНОЙ ГАВАНИ (SAFE HARBOR)

Это обязательный юридический слайд.

Всё содержимое этого доклада может быть художественным вымыслом или неправдой. Поэтому, нужно думать своей головой и не верить докладчику на слово.

Если вы пытаетесь внедрить решения и знания, полученные из этого доклада, **вам стоит нанять профессионалов**. Они расскажут, что стоит использовать, а что — нет.

## 2025 год. Цифры.

- **$202B** — инвестиции в AI за год
- **800M** еженедельных пользователей ChatGPT
- **50%** всего венчурного капитала мира — в AI
- Брет Тейлор (OpenAI): «У нас тоже пузырь»

Кассовый вопрос: **заменит ли AI-чат программистов?**

Часть первая

# Нужны ли вообще чаты?

# 80 лет битвы чатов с файлами

## Этот вопрос задавали раньше

1961 — «Интерактивные терминалы заменят пакетную обработку!»
1973 — «Живая среда LISP заменит файлы!»
1985 — «Экспертные системы заменят специалистов!»
1997 — «UML-диаграммы заменят код!»
2022 — «ChatGPT заменит программистов!»

Каждый раз — **файлы и работа людей над ними побеждали.**

# Цикл 1

# Первый «чат» в истории

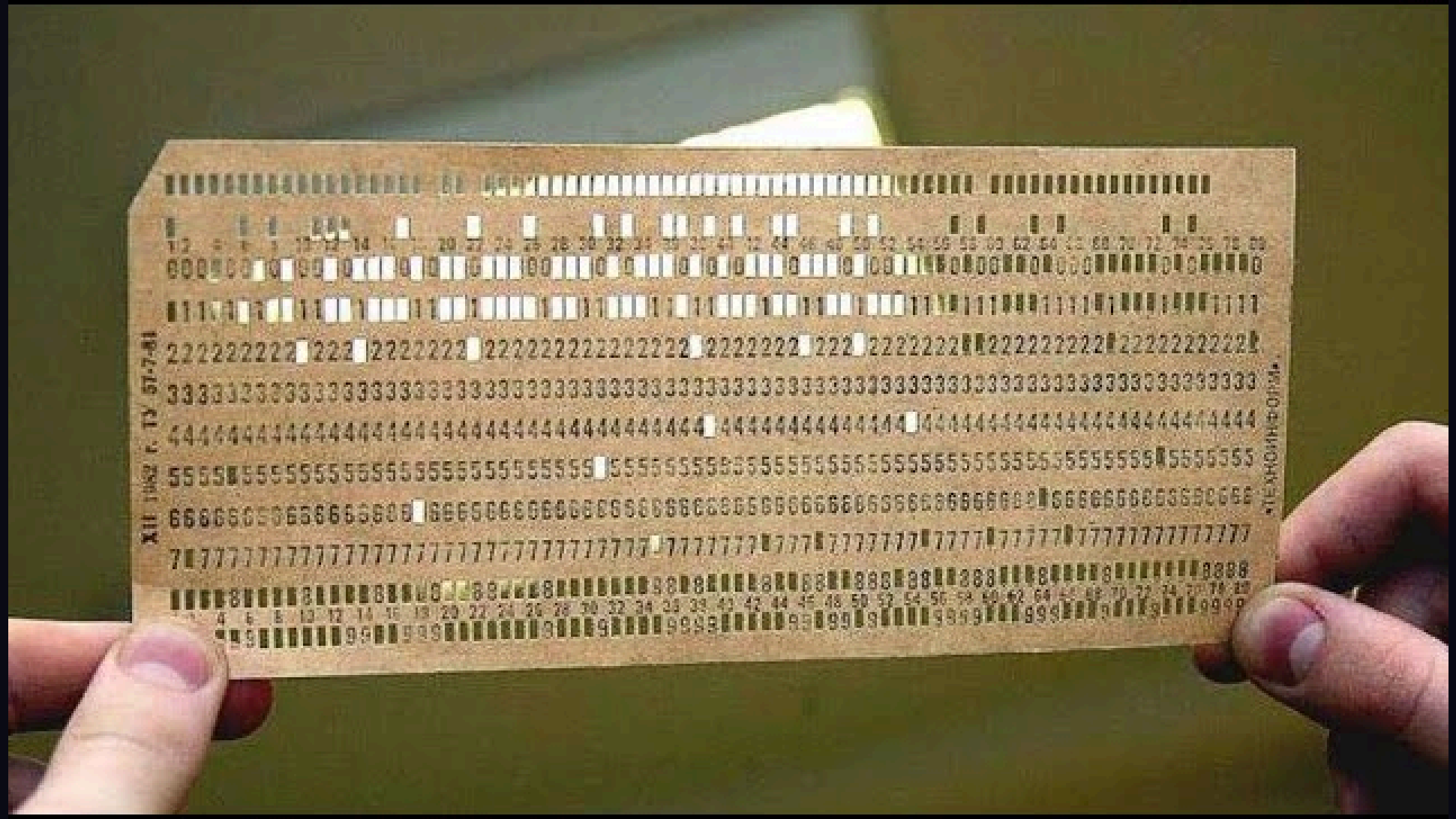
1961 — 1972

# До 1961: компьютер размером с дом

- ENIAC (1945): 27 тонн, 174 кВт
- Программирование = **перекоммутация проводов**
- Перфокарты: пробил → сдал оператору → **ждёшь часы**
- Ошибка = повторить весь цикл

Программист **не видит** компьютер.


GE-645 SYSTEM



# 1961: CTSS — первый диалоговый интерфейс

**Fernando Corbató, MIT. IBM 7094.**

Несколько пользователей одновременно.
Печатаешь команду → получаешь ответ. **Сразу.**

> *«Man-Computer Symbiosis»*
> *— Licklider, 1960*

Интерактивные пользователи работали **на порядок** продуктивнее.


TimeShare
Corporation
START
STOP
FREE


COMPILERS TOOK MY JOB
imgflip.com

# 1966: ELIZA — первый чат-бот

**Joseph Weizenbaum, MIT. ~200 правил.**

Программа прикидывается психотерапевтом.
Чистый pattern-matching. Никакого «понимания».

Но секретарь Вейценбаума попросила его
**выйти из комнаты** —
чтобы поговорить с ELIZA приватно.

(Возможно, анекдот — но показательный.)

# E.L.I.Z.A. Talking

# Развязка: Unix, C и «всё есть файл»

Thompson и Ritchie уходят из Multics.
Создают Unix на PDP-7.

Философия: **«всё есть файл»**

- `ed` (1969) → `vi` (1976) — файловые редакторы
- **C** (1972) — файловый язык: редактируй → компилируй → запускай
- Pipe, текстовые потоки, конфиг-файлы

Файловый подход определил **следующие 50 лет.**

# Цикл 2

# Будущее, которое не случилось

1973 — 1983

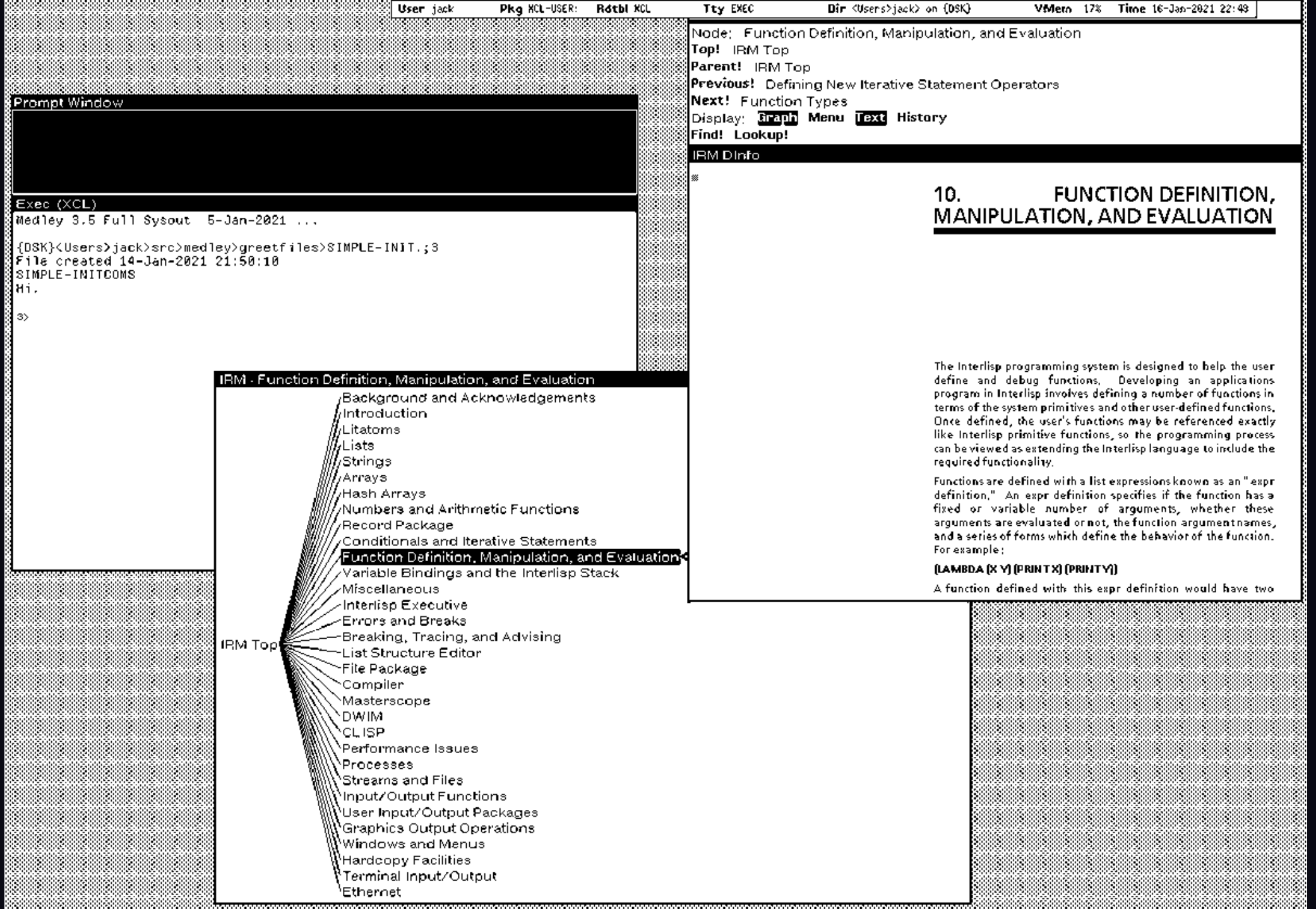
# Interlisp-D: IDE из будущего

**Xerox PARC. Teitelman & Bobrow.**

- **DWIM** — Do What I Mean (автоисправление)
- Structure-aware editing
- Live debugging — отладка прямо в работающей программе
- Программа — не файл, а **живой объект** в памяти

В некоторых аспектах — продвинутее Claude Code 2025 года.

Нечто подобное: JetBrains MPS (используется для исследований и на заводах)


User jack Pkg XCL-USER: Rdtbl XCL Tty EXEC Dir <Users>jack> on {DSK} VMem 17% Time 16-Jan-2021 22:49
Prompt Window
Exec (XCL)
Medley 3.5 Full Sysout 5-Jan-2021 ...
{DSK}<Users>jack>src>medley>greetfiles>SIMPLE-INIT.;3
File created 14-Jan-2021 21:50:10
SIMPLE-INITCOMS
Hi.
3>
Node: Function Definition, Manipulation, and Evaluation
Top! IRM Top
Parent! IRM Top
Previous! Defining New Iterative Statement Operators
Next! Function Types
Display: Graph Menu Text History
Find! Lookup!
IRM DInfo
10. FUNCTION DEFINITION, MANIPULATION, AND EVALUATION
The Interlisp programming system is designed to help the user define and debug functions. Developing an applications program in Interlisp involves defining a number of functions in terms of the system primitives and other user-defined functions. Once defined, the user's functions may be referenced exactly like Interlisp primitive functions, so the programming process can be viewed as extending the Interlisp language to include the required functionality.
Functions are defined with a list expressions known as an "expr definition." An expr definition specifies if the function has a fixed or variable number of arguments, whether these arguments are evaluated or not, the function argument names, and a series of forms which define the behavior of the function. For example:
(LAMBDA (X Y) (PRINTX) (PRINTY))
A function defined with this expr definition would have two
IRM - Function Definition, Manipulation, and Evaluation
IRM Top
Background and Acknowledgements
Introduction
Litatoms
Lists
Strings
Arrays
Hash Arrays
Numbers and Arithmetic Functions
Record Package
Conditionals and Iterative Statements
Function Definition, Manipulation, and Evaluation
Variable Bindings and the Interlisp Stack
Miscellaneous
Interlisp Executive
Errors and Breaks
Breaking, Tracing, and Advising
List Structure Editor
File Package
Compiler
Masterscope
DWIM
CLISP
Performance Issues
Processes
Streams and Files
Input/Output Functions
User Input/Output Packages
Graphics Output Operations
Windows and Menus
Hardcopy Facilities
Terminal Input/Output
Ethernet


✻ Welcome to the Claude Code research preview!
CLAUDE CODE
🎉 Login successful. Press Enter to continue

# Smalltalk: вся программа — живая

**Alan Kay, Dan Ingalls. Xerox Alto.**

Image-based: вся программа — живой образ в памяти.
Нет файлов. Нет компиляции. Всё — **прямо сейчас.**

Вдохновил Macintosh.

Но: нет контроля версий, нет совместной работы.

Fullscreen

Classes

| | | | |
|---|---|---|---|
| AllClasses | CharLine | ClassDefinition | complementfrom: |
| SystemOrganization | Dispframe | ClassOrganization | compRect: |
| 'Kernel Classes' | Font | 'all' | reversefrom:to: |
| 'Numbers' | FontSet | 'start & finish' | selection |
| 'Basic Data Structures' | Paragraph | 'menu messages' | |
| 'Sets and Dictionaries' | RemoteParagraph | 'accessing' | |
| 'Graphical Objects' | StyleSheet | 'coloring' | |
| 'Text Objects' | Textframe | 'editing keys' | |
| 'Windows' | Textlmage | 'selecting' | |
| 'Panes and Menus' | | 'character shapes' | |
| 'Compiler' | | 'indicating' | |
| 'NoteTaker Classes' | | 'copying' | |

```
compRect: r
	["Reverse the rectangle r from white to black as part of a selection"
	((r intersect: frame) intersect: window) comp.
	(( (r copy translate: 2@2) intersect: frame) intersect: window) comp.]
```

UserView.workspace

XEROX - Learning Research Group

This is a resurrected version of the Smalltalk-78 system running on the Notetaker computer in 1979.

tended from the original, mainly
gs (the Notetaker never went
no), and by restoring features
k-76 that were stripped out to
taker's limited hardware. Other
been added to take advantage of
dern machine speeds.

Enjoy!
Dan, Ted, Yoshiki, Alan
2014

100 factorial
factorial/99 factorial

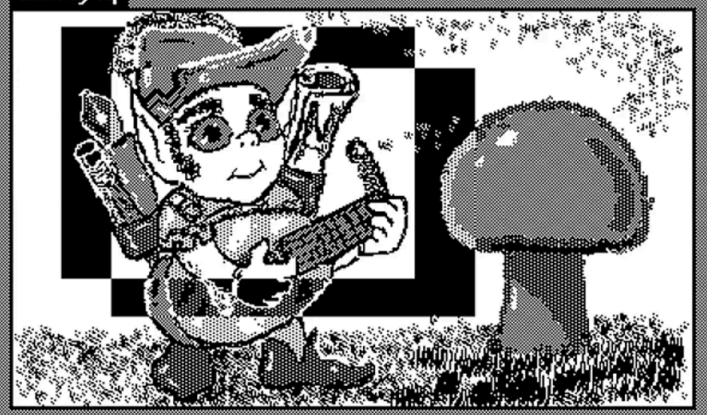
The Jobs Demo

To recreate my change to text selection, first execute "Rectangle fromuser comp." a few times to see how simple black/white complement works.

Then browse to
'Text Objects'/Textimage/'selecting'/complementfrom:to: and observe that it calls compRect: to complement the three parts of any selection.

Then browse to compRect:, and below the line:
((r intersect: frame) intersect: window) comp.
add another that complements a second offset rectangle:
(( (r copy translate: 1@1) intersect: frame) intersect: window) comp.
The circle dot is infix to make an x-y point; it's typed as an at-sign.
You can just copy the whole line from here.
Choose 'compile' from the edit menu - poof - selections are now outlined!

Change the offset to 2@2 to look even better. Note how long it takes to recompile this change to the heart of the system.



Reset ■ | Load image | Go to 0 | Step ▷ | Running ◐ | Free oops: 18014 | Help | File...

error

# Развязка: $100K vs $49.99

| LISP-машина | IBM PC + Turbo Pascal |
|---|---|
| $100 000 | $1 565 + $49.99 |
| Специализированная | Универсальная |
| Живая среда | Edit → compile → run |
| ~10 000 продано | **600 000** за 3 года |

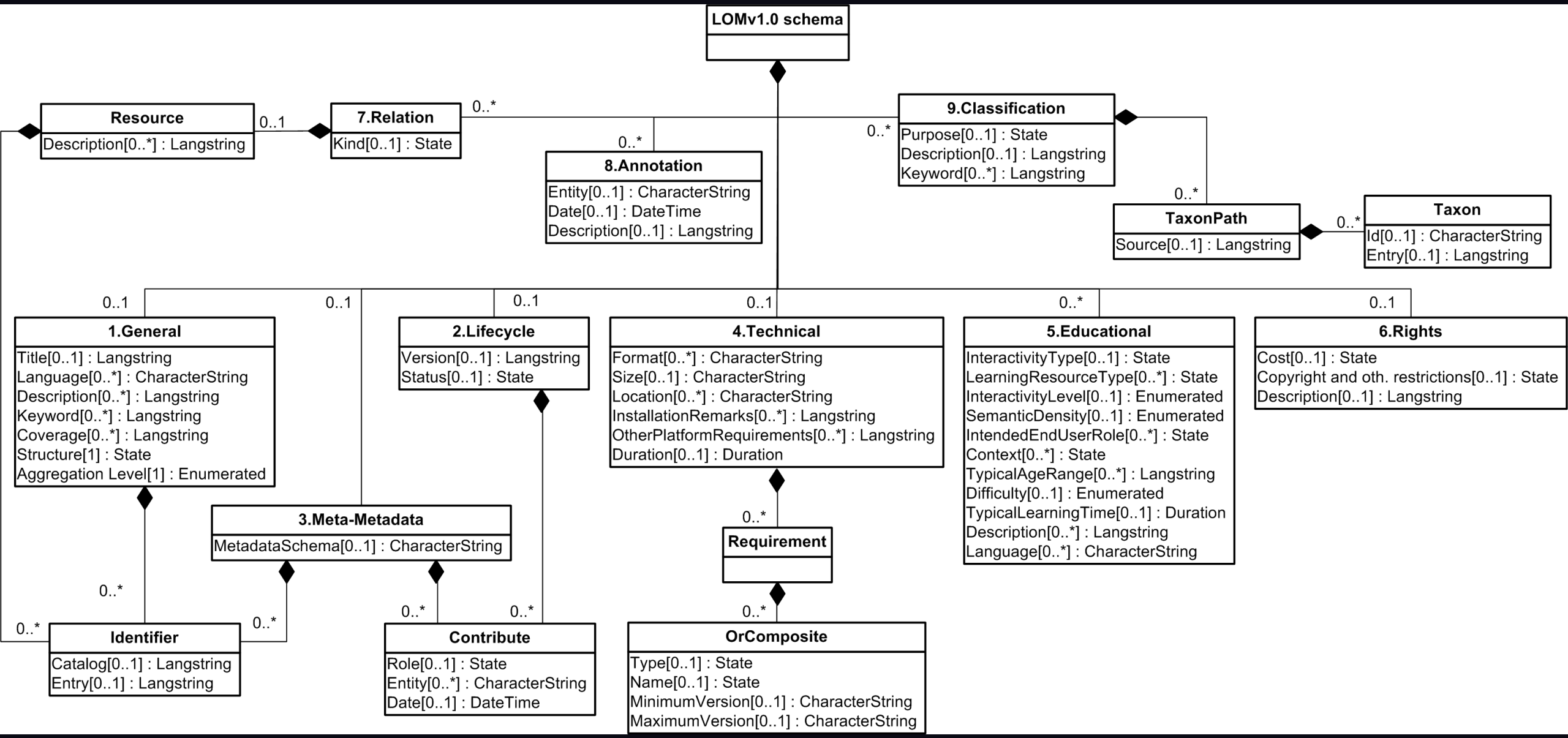
Будущее проиграло настоящему.

**Файлы: 2, Чат: 0.**

# Цикл 3

# «Заменим программистов диаграммами»

UML · 1997 — 2016


LOMv1.0 schema
Resource
Description[0..*] : Langstring
0..1
7.Relation
Kind[0..1] : State
0..*
8.Annotation
Entity[0..1] : CharacterString
Date[0..1] : DateTime
Description[0..1] : Langstring
0..*
9.Classification
Purpose[0..1] : State
Description[0..1] : Langstring
Keyword[0..*] : Langstring
0..*
TaxonPath
Source[0..1] : Langstring
0..*
Taxon
Id[0..1] : CharacterString
Entry[0..1] : Langstring
0..1
1.General
Title[0..1] : Langstring
Language[0..*] : CharacterString
Description[0..*] : Langstring
Keyword[0..*] : Langstring
Coverage[0..*] : Langstring
Structure[1] : State
Aggregation Level[1] : Enumerated
0..1
2.Lifecycle
Version[0..1] : Langstring
Status[0..1] : State
0..1
4.Technical
Format[0..*] : CharacterString
Size[0..1] : CharacterString
Location[0..*] : CharacterString
InstallationRemarks[0..*] : Langstring
OtherPlatformRequirements[0..*] : Langstring
Duration[0..1] : Duration
0..*
5.Educational
InteractivityType[0..1] : State
LearningResourceType[0..*] : State
InteractivityLevel[0..1] : Enumerated
SemanticDensity[0..1] : Enumerated
IntendedEndUserRole[0..*] : State
Context[0..*] : State
TypicalAgeRange[0..*] : Langstring
Difficulty[0..1] : Enumerated
TypicalLearningTime[0..1] : Duration
Description[0..*] : Langstring
Language[0..*] : CharacterString
0..1
6.Rights
Cost[0..1] : State
Copyright and oth. restrictions[0..1] : State
Description[0..1] : Langstring
0..1
3.Meta-Metadata
MetadataSchema[0..1] : CharacterString
0..*
Requirement
0..*
Identifier
Catalog[0..1] : Langstring
Entry[0..1] : Langstring
0..*
Contribute
Role[0..1] : State
Entity[0..*] : CharacterString
Date[0..1] : DateTime
0..*
OrComposite
Type[0..1] : State
Name[0..1] : State
MinimumVersion[0..1] : CharacterString
MaximumVersion[0..1] : CharacterString

# UML + MDA: «модель это код»

**IBM купил Rational за $2.1B** (2003)

- 14 типов диаграмм
- Обещание: рисуешь диаграмму → получаешь код
- Освоить UML **сложнее**, чем написать код, который он призван заменить

Сгенерированный код — раздутый, неидиоматичный, неподдерживаемый.

## Agile убивает UML

**2001, Snowbird, Юта. 17 практиков.**

> *«Работающий софт важнее исчерпывающей документации.»*

Martin Fowler — автор UML Distilled — **среди подписавших.**

2016: Visual Studio убирает поддержку UML.

# Черное зеркало

| UML (1997) | AI Chat (2022) |
| --- | --- |
| Заменим код **диаграммами** | Заменим код **промптами** |
| Модель это код | Промпт это код |
| Сгенерированный код — неподдерживаемый | Сгенерированный код — ? |
| Последняя миля — код пишется вручную | Последняя миля — код пишется вручную? |

Оба недооценивают **сложность реального софта.**

# Цикл 4

# Отражение 2025 года

Экспертные системы · 1981 — 1993


BACKWARD CHAINING

GOAL: Make $20.00

RULE: If the lawn is shaggy and
the car is dirty and you mow
the lawn and wash the car,
then Dad will give you $20.00

Does the lawn need mowing?

Do you have a mower?

gas? electric? push?

Does the car need washing?

hose? bucket? rags?

*** The inference engine will test each rule or ask the user for additional information.

## Экспертные системы: хайп 1980-х

- DENDRAL, MYCIN, R1/XCON
- DEC экономит **$40M/год** с R1/XCON
- **$1B+/год** инвестиций к 1985
- «Новые AI-компании — по одной в неделю»
- Япония: Fifth Generation Computer, **$850M**

## Крах за один год

**1987:** Рынок LISP-машин **уничтожен за один год.**

- 300+ AI-компаний закрылось
- Symbolics — банкрот
- Шенк и Минский предупреждали ещё в 1984

Джек Шварц (DARPA):

> *«Это просто хитрое программирование.»*

# Параллели

| | 1985 | 2025 |
|---|---|---|
| Инвестиции | $1B+/год | **$200B+/год** |
| Обещание | AI заменит экспертов | AI заменит программистов |
| Реальный результат | Экономия в нишах | Экономия в нишах |
| Хрупкость | Не обобщает | Галлюцинации |
| Доверие ↓ | Минский: «хайп» | SO: 70% → 60% |

# Между циклами

# Три победы файлов, о которых забывают

# Три «тихие» победы

**GUI** (1984 → 1995)
Macintosh, Windows 95. Самая массовая интерактивная революция.
Но метафора рабочего стола — это **папки и файлы.**
Save As. Drag-and-drop. GUI = визуализация файлов.
Cortana Chat в Windows 11 не взлетел.

**Веб** (1993 → сегодня)
Четвёртый великий интерфейс. HTML/CSS/JS — файлы.
URL — адрес ресурса. HTTP GET — «дай файл».
В Chromium Developer Tools экспериментируют, а не пишут в прод.

**Jupyter** (Python, 2014) и **Zeppelin** (Scala, 2015)
10M Jupyter-ноутбуков на GitHub. Но стандартный workflow:
«Исследуй в notebook → перенеси в .py файлы.»
Очень медленный прогресс.

# Текущий цикл

# AI-революция в реальном времени

2022 — 2025


ChatGPT: Optimizing Language Models for Dialogue
vodafone P
12:01
31%
openai.com
Try ChatGPT ↗
Learn more ›
OpenAI
API
RESEARCH
BLOG
ABOUT
ChatGPT: Optimizing Language Models for Dialogue
We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and

Directed by
ROBERT B. WEIDE

# ChatGPT: восьмой «чат»

**Ноябрь 2022.** 100M пользователей за 2 месяца.

Восьмая итерация **того же паттерна:**

CTSS → ELIZA → MS-DOS → VB/Delphi →
→ Jupyter → ChatGPT

Новейший «чатовый» интерфейс computing.
А дальше?

# Файлы контратакуют

**Copilot** (2021) → **Cursor** (2023)
AI интегрирован в файловые IDE.
Cursor: $1B ARR. AI-first, но **файловый.**

**Claude Code** / **Codex CLI**
Терминальные агенты. Читают и пишут **файлы.**
`CLAUDE.md` — конфиг через markdown-файл.

**Kiro** (AWS, 2025)
`requirements.md → design.md → tasks.md → код`
AI-революция породила инструмент,
генерирующий **спецификации в файлах.**

# Git — финальный аргумент

**Linus Torvalds, 2005. Distributed VCS.**

`diff` — только для файлов
`merge` — только для файлов
`branch` — только для файлов
`code review` — только для файлов

Git не работает с чат-сессиями,
REPL-историями или живыми образами.

**Контроль версий — структурное преимущество файлов.**

## Git не работает с чат-сессиями?

## А если найду?

**Thomas Dohmke, стартап EntireHQ**

- $60 млн в сид раунде с оценкой в $300 млн
- AI-first Git Platform
- Checkpoints: автоматическое сохранение контекста агентов в Git

Всё это тоже файлы, хоть и гибридные

# Почему файлы побеждают. Каждый раз.

**1. Контроль версий**

diff, merge, branch, blame — только текстовые файлы

**2. Совместная работа**

Code review, pull requests, CI/CD — только файлы

**3. Воспроизводимость**

Файл на моей машине = файл на вашей машине = файл на сервере

Эти свойства **не зависят от технологии.**

Они работали в 1972. Работают в 2025.

# 8 циклов. Один паттерн.

1961 Time-sharing → Unix: «всё есть файл»
1973 LISP-машины → IBM PC + Turbo Pascal
1984 GUI-революция → Рабочий стол = папки + файлы
1985 Expert Systems → Крах за один год
1993 Веб-браузер → HTML/CSS/JS = файлы
1997 UML: модель→код → Agile: «работающий софт»
2014 Jupyter → «Исследуй → перенеси в .py»
2022 ChatGPT → Cursor, Claude Code, Kiro

## Вопрос не «чат или файлы?»

Интерактивные инструменты **не умирают.**
REPL жив. Jupyter жив. ChatGPT будет жить.

Вопрос: **сколько времени мы используем чат, и сколько файлы?**

Исследуй в чате.
Разрабатывай в файлах.

Это не что-то новое, это **80 лет эволюции.**

Часть вторая

# Новое лето искусственного интеллекта

# Как мы дошли до 100% генерации кода

# Hardware → Science → Products → Practices

Пять волн AI-революции в разработке

Февраль 2026

# Каскад: от железа до нашей повседневной работы

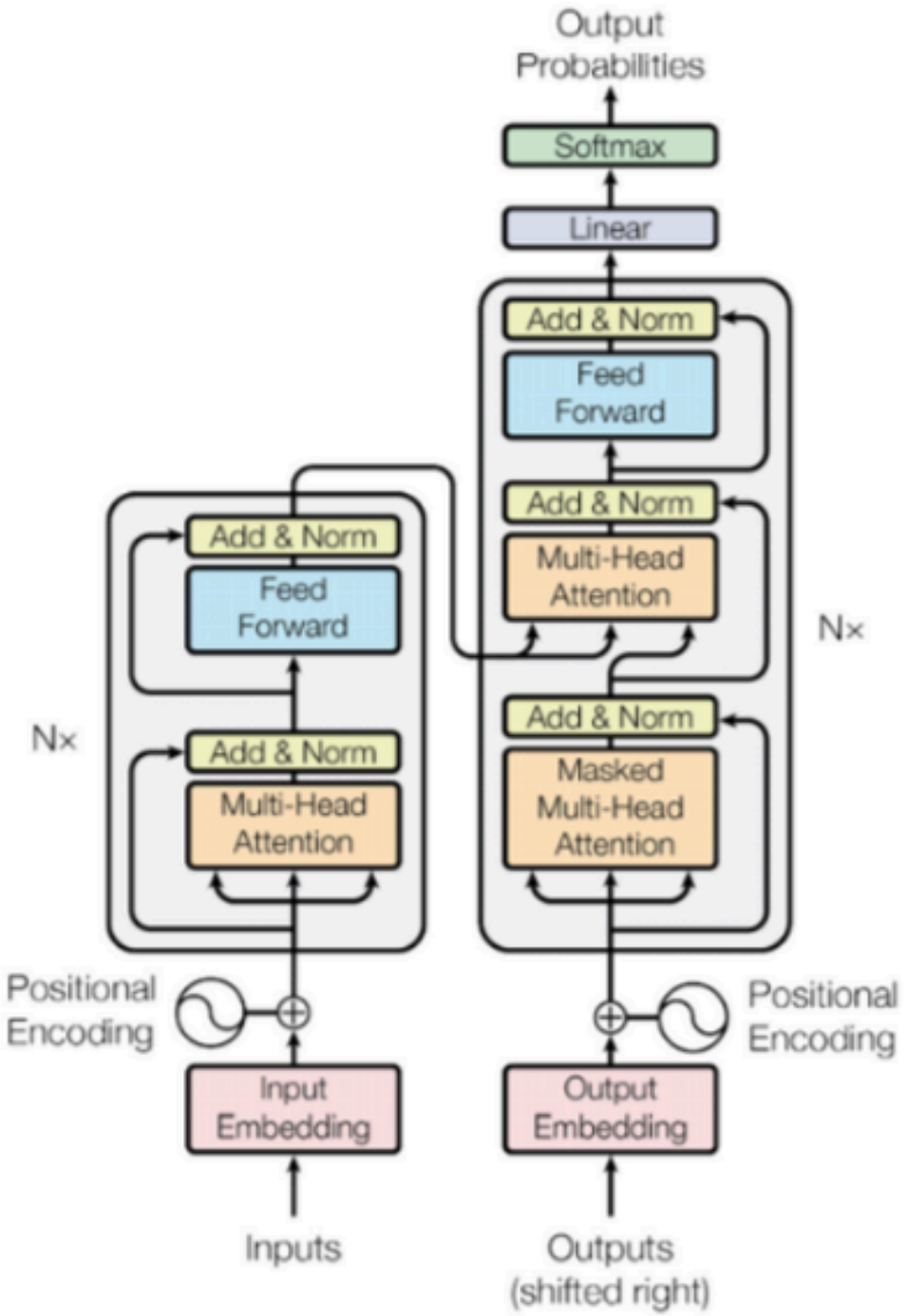
- **Hardware** — железо определяет границы возможного
- **Science** — учёные осваивают новые горизонты
- **Products** — то, что вы можете скачать и запустить
- **Practices** — как вы реально изменили подход к работе

> *Задержка каскада:*
> ***4 года** в 2017 → **6 месяцев** в 2025.*
> *Волны ускоряются и **перекрываются**.*


NVIDIA
TR21
1704

# Transformer

## Attention Is All You Need



1

# Волна Tensor Cores

2017 → 2021. Четыре года от чипа до продукта

V100 → Transformer + RLHF → Copilot → Comment-Driven Dev

Самая длинная волна. Четыре года тишины между открытием и продуктом.

## V100: железо, которое зажгло искру

**125 TFLOPS** для deep learning. Tensor Cores: матричные операции **5–12× быстрее**

## Transformer: архитектура, которая победила всех

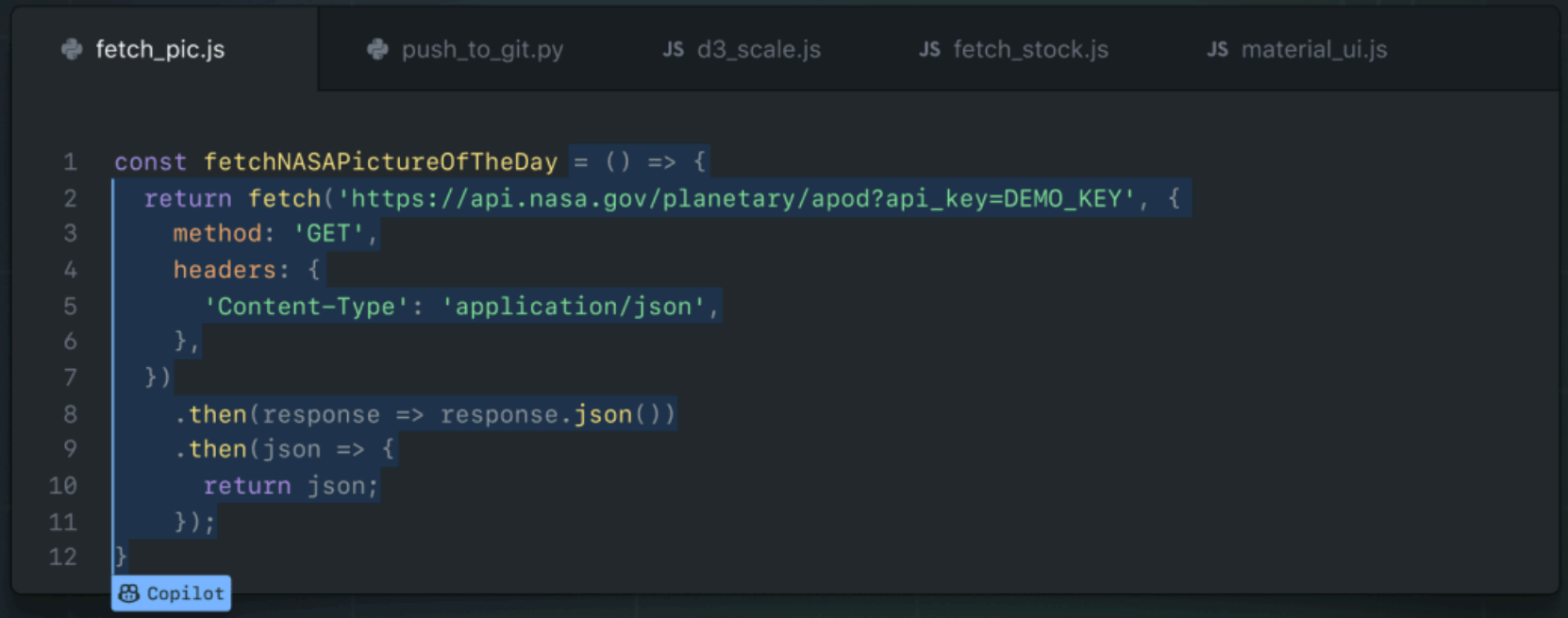
Self-attention заменяет рекуррентность.
**Параллельная** обработка всей последовательности. **O(1)** между любыми позициями

В том же году — RLHF foundations (Christiano et al.). Семя, которое прорастёт через 5 лет

> *Без Tensor Cores Transformer остался бы академическим курьёзом.*
> *Это **первый** пример каскада: чип делает науку возможной.*

Technical preview

# Your AI pair programmer




GitHub Copilot

## Четыре года тишины → Copilot

**2018–2020**: ученые работает, продуктов нет.

- BERT (2018), GPT-2 (2019), GPT-3 (2020)
- Программисты ещё не заметили

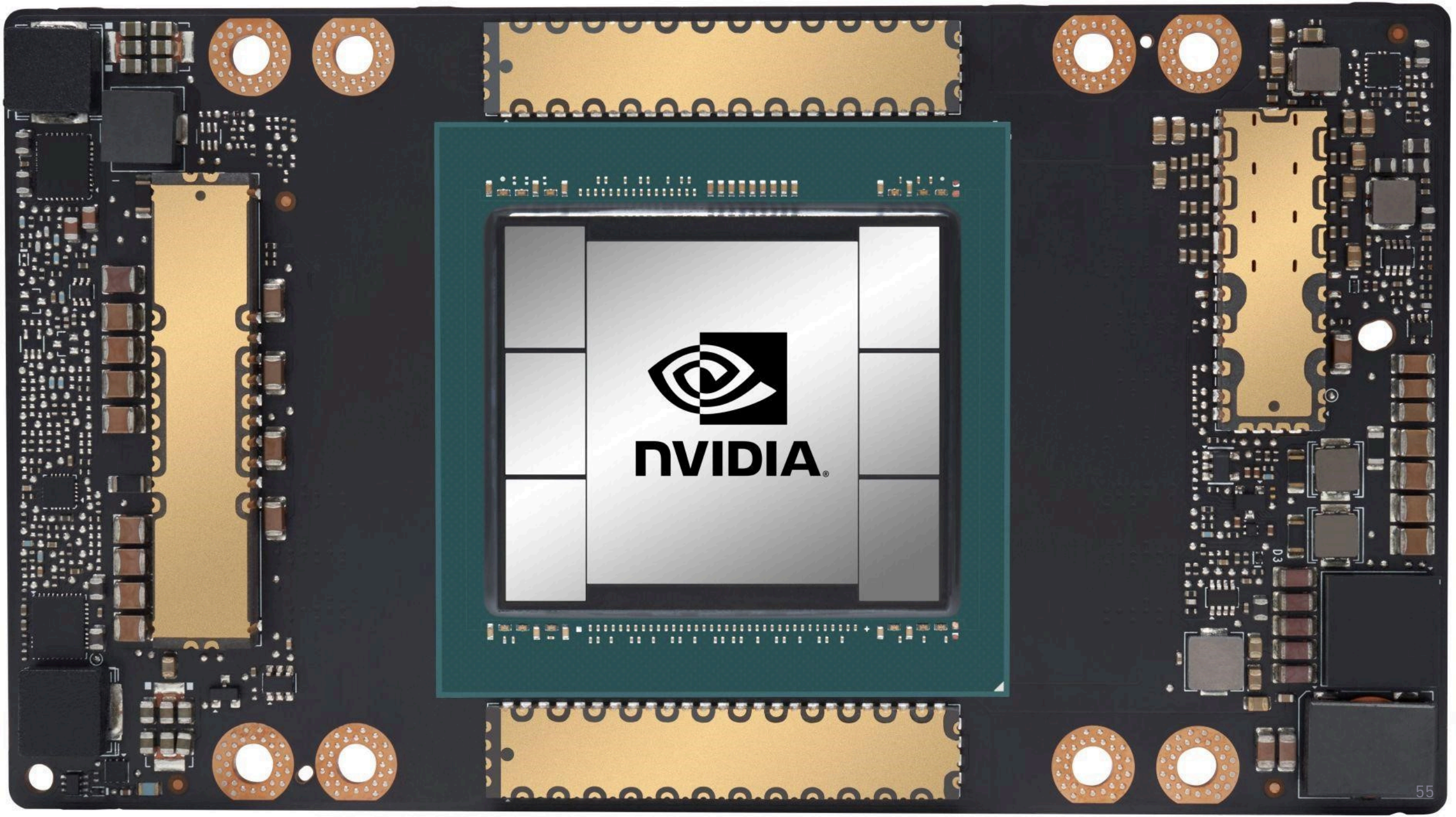
**Июнь 2021 — GitHub Copilot Preview**
Codex (GPT-3, дообученный на коде). **Первый массовый AI-ассистент для кода**

**27%** кода в файлах генерирует AI. **1M+** пользователей за год

Новая практика: Comment-Driven Development

Комментарии стали **входом**, а не выходом. Напиши что — AI напишет как

> Задержка волны 1: **4 года** от железа до продукта


NVIDIA

2

# Волна Масштаба

2020 → 2023. Больше = лучше (пока не перестало)

A100 → GPT-3, Scaling Laws, CoT, InstructGPT → ChatGPT → SO умирает, Prompt Eng, RAG

**Scaling is all you need**

Волна, в которой масштаб породил эмерджентность
И парадокс: маленькая модель + RLHF лучше, чем огромная модель

## A100 → Scaling Laws + парадокс InstructGPT

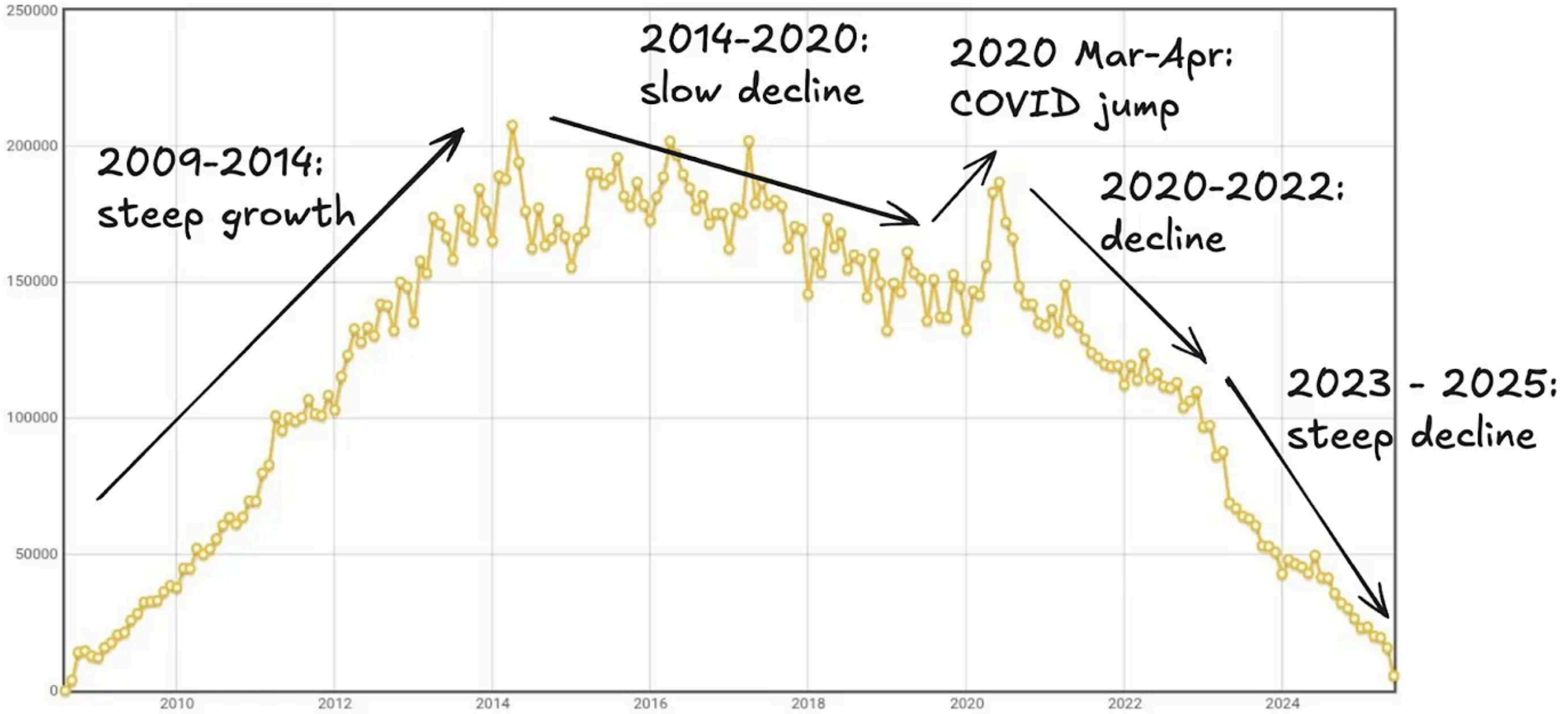
**A100**: 80 ГБ HBM2e, 2 ТБ/с. **GPT-3 (175B)** тренировался на кластерах A100

**Scaling Laws** (Kaplan et al.): loss = степенная функция от параметров, данных, compute
«Закон Мура для AI»: предсказуемая отдача при 10× увеличении

### Парадокс 2022: маленькая побеждает большую

**InstructGPT**: 1,3B с RLHF **побеждает** 175B GPT-3 без
**Chain-of-Thought** (Wei et al.): «Let's think step by step» — **кратно** улучшает точность

> *Pre-training scaling работает, но **post-training определяет полезность**.*
> *H100 с нативным FP8 (**4× throughput** vs A100) делает это масштабируемым.*




pragmaticengineer.com

## ■ ChatGPT → ■ Stack Overflow умирает за 24 месяца

**30 ноября 2022** — ChatGPT. Самая быстрая adoption в истории

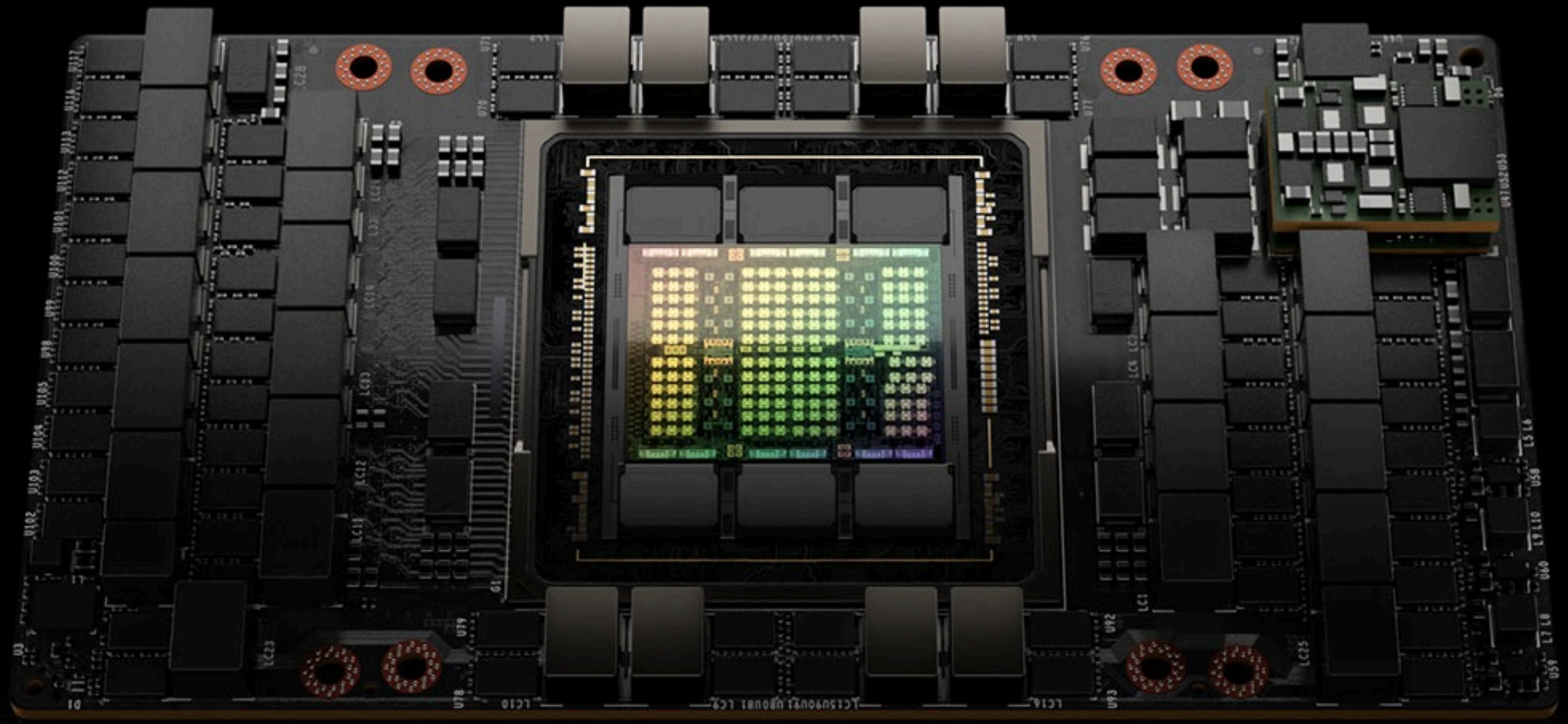
### Stack Overflow

- ноябрь 2022: **108 563** вопросов/мес
- январь 2023: **96 377** (-11%)
- декабрь 2024: **25 566** (-76,5%)

PNAS Nexus: **-25% каузальное** снижение. SO: увольнения **10%** + **28%**

### Одновременно — рождение трёх практик

**Prompt Engineering**: Indeed: 2 → **144** вакансии/млн за 3 месяца
**QLoRA**: 65B модель на **одном 48GB GPU** — «уравниватель»
**RAG**: **51%** enterprise AI. Pinecone $100M, Weaviate $50M



3

# Волна Эффективности

2022 → 2025. Не «больше» — а «умнее при тех же ресурсах»

H100 FP8, MI300X, MXFP4 → MoE, Mamba, KV cache → Cursor, AI IDEs → AI-native IDE, «write less»

Волна, где MoE стала доминирующей архитектурой, а Cursor — самым быстрым SaaS в истории

## Мощное и эффективное железо

**H100 FP8** (2022): обучение и inference в пониженной точности **без деградации**. 4× throughput
**MI300X** (2023): **192 ГБ** единой памяти — AMD конкурирует с NVIDIA
**MXFP4** (2025): 4-битная квантизация. **120B на одном H100** (90% MMLU)

## MoE: архитектурный ответ на проблему масштаба

Mixtral 8×7B: из ~47B параметров **работают ~13B на токен**
**Качество большой модели при стоимости маленькой**

К 2025: Qwen3-235B (128 экспертов, top-8), DeepSeek-V3, Grok-3
ResMoE: **75% сжатие**. **Каждая frontier-модель 2025 — MoE**

Параллельно: Mamba (SSMs) — O(n) вместо $O(n^2)$. Единственный реальный конкурент Transformer, но для кода пока не победил

# Cursor: самый быстрорастущий SaaS в истории

**AI-native IDE** — не «плагин в IDE», а **«AI с IDE вокруг»**

$1M ARR → $100M ARR за **~12 месяцев**. К 2026: **$29,3B** валюация, **$1B+** ARR
**1 миллиард** строк принятого кода **в день**. Ни одного доллара на маркетинг

## Cursor 2.0

Composer model (MoE + RL)
**4× быстрее**, ~250 т/с
До **8 параллельных** агентов

## Под капотом

RAG + **Merkle tree** для изменений
Turbopuffer vector DB
**Сотни ТБ** эмбеддингов

> *Jensen Huang назвал Cursor **«my favorite enterprise AI service»***

# Практики волны эффективности

## «Пиши меньше, ревьюй больше»

Google: **>25% кода** AI-сгенерировано (Pichai, Q3 2024)
**46%** кода у активных Copilot-юзеров — AI. Java-разработчики: **61%**

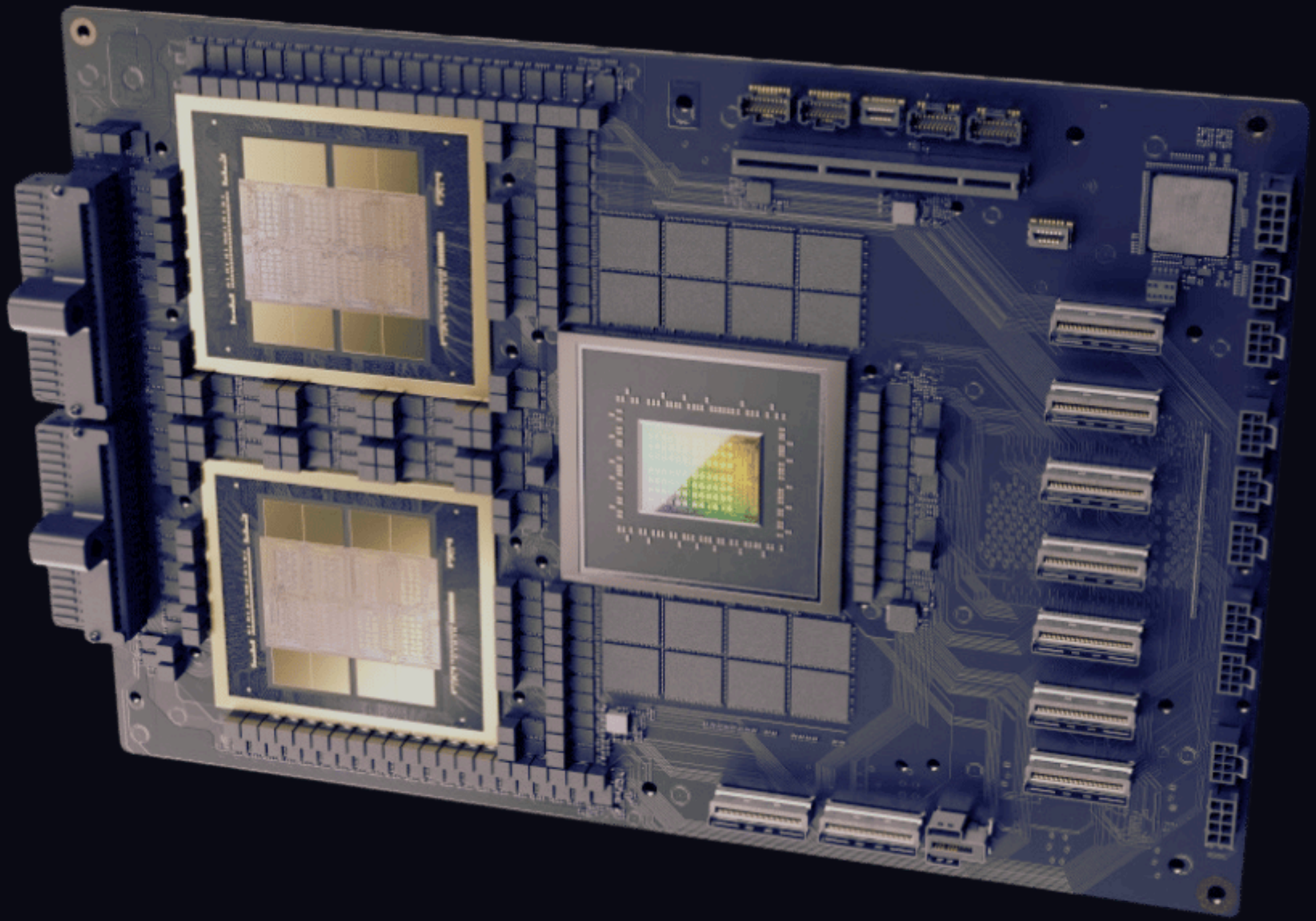
Werner Vogels: **«Verification debt»** — ревью AI-кода **сложнее**, чем своего

## AI-native IDE как категория

Cursor + Windsurf → десятки тысяч переходят. **85%** используют AI daily (JetBrains 2025)

## MCP — «USB-C для AI» (ноябрь 2024)

Anthropic выпускает открытый стандарт. JSON-RPC, вдохновлён LSP.
Пока мало кто заметил. Но через год — 97M downloads



4

# Волна Reasoning

2024 → 2026. «Думай дольше, а не больше»

B200 FP4, Groq, TPU v6 → TTC, GRPO ★, DPO → o1, R1, Claude Code, Codex → Vibe Coding, Agents, AI Review

Волна, которая перевернула парадигму: reasoning — emergent property из чистого RL

## Железо эры инференса

**B200**: нативный FP4, 192 ГБ HBM3e, **~2.5× inference/GPU** vs H100
**Groq LPU**: только inference. SRAM, детерминизм. **~300 т/с** Llama 70B
**TPU v6**: **4,7×** vs v5e. Google работает на своём железе

> *Inference-железо ≠ training-железо.*
> *Конец эпохи «один GPU для всего».*
> NVIDIA купила Groq за $20B (2025)

## Test-Time Compute: переворот

Llemma-**7B** + tree search **>** Llemma-**34B** на MATH
«Дай модели подумать» вместо «сделай модель больше»

**DPO**: убирает reward model → alignment для **команд без тысяч своих GPU**

## ★ GRPO: reasoning возникает из ничего

**DeepSeek-R1-Zero** — чистый RL, **без SFT**, без примеров рассуждений
Только сигнал «правильно/неправильно». И reasoning **возник сам**

AIME 2024: **15,6% → 71,0%** pass@1. **Nature**, сентябрь 2025

Что появилось без программирования:

- Self-reflection — модель проверяет свои шаги
- Verification — модель ищет ошибки в своём ответе
- **«Aha moments»** — модель поправляет себя mid-reasoning

> *Открытие уровня **«Attention Is All You Need»**.*
> *Раньше: нужны человеческие примеры рассуждений.*
> *Теперь: дай сигнал «верно/неверно» — reasoning **возникнет сам**.*

# Два подхода к агентному кодированию

### Claude Code

Терминальный, **in-the-loop**
72,7% → **80,9%** SWE-bench (Opus 4.5)
90% своего кода пишет сам
**$1B** ARR

### Codex (OpenAI)

Облачный, **async**
**56,8%** SWE-bench Pro (SOTA)
**77,3%** Terminal-Bench 2.0
10 задач параллельно

**o1** (2024): первая reasoning-модель в проде. 30 сек → правильный ответ
**DeepSeek-R1** (open-weight): 7B = 55,5% AIME. **Каждый может запустить**

> *Copilot: **20M+** юзеров, **42%** рынка, **1.2M PR/мес** через coding agent*

# VIBE CODING

# Vibe Coding + Background Agents

> «There's a new kind of coding I call 'vibe coding', where you fully give in to the vibes, embrace exponentials, and forget that the code even exists.»
> — ***Andrej Karpathy***, *2 февраля 2025.* ***Collins Word of the Year***

Y Combinator W25: **25%** стартапов — 95% AI-код
Но **72% профи: «это не мой метод»**

## Агентное программирование: делегируй фичу

«AI реализует фичи, **пока я занят другим**»
Задача → sandbox → PR. Claude Code + Codex + Copilot Agent

## AI Code Review: 14% → 51% за 10 месяцев

AI в **1/7 PR**. Qodo: **81%** улучшение качества

5

# Волна Спецификаций

2025 → ... Только начинается

NVFP4 → Astrogator, Free Transformer, EvoMAC → Kiro, Antigravity, Verified GenCode → Spec-Driven Dev, MCP стандарт, Teams Shrink

Волна, которая отвечает на вопрос: «как доверять AI-коду?»

# Железо и наука о верификации

**NVFP4**: 4-bit KV-кеш. **50%** сокращение памяти → **50% больше контекста** на GPU
QuantSpec + ShadowKV + MagicDec → **128K контекст управляем**

## Наука верификации

| Подход | Результат | Вердикт |
|---|---|---|
| **Astrogator** (domain-specific) | **83%** верификация, **92%** детекция | Работает! |
| CoT+RAG (general) | 100% спеки, **58%** верификация | Недостаточно |
| ~~Full auto-formalization~~ | 58% — непригодно | Тупик |

**Free Transformer** (Meta): планирует **ДО** генерации. +11% код на 8B
**EvoMAC**: агенты **перестраивают свою топологию**. +26,5% на Website

> *Путь к верифицированному коду — **domain-specific**, не general*

# Продукты волны спецификаций

## Kiro (AWS)

**Spec-driven development**

requirements.md → design.md → tasks.md → code

Контроль через спецификации, не промпты

## Antigravity (Google)

**Agent-first IDE**

Manager View: **5+ параллельных** агентов

Артефакты + скриншоты как доказательства

SWE-BENCH VERIFIED

2023: **2%** → 2025: **75%** 37× за два года

kiro.app

requirements.md U | design.md U | tasks.md U

.kiro > specs > email-opt-in > tasks.md > implementation plan

Spec: email-opt-in | 1 Requirements | 2 Design | 3 Task List

```
1  # Implementation Plan
2
   ✓ Task completed
3  [x] 1. Create the EmailOptInForm component
4  - Create a new React component for the email opt-in form with
   basic structure
   - Implement form UI with input field and submit button
   - Add Tailwind CSS styling to match the application design
   - _Requirements: 1.1, 1.2, 1.4_

5    Start task
6  [ ] 2. Implement form validation and state management

7    Task in progress
8  [ ] 2.1 Add form state management using React hooks
9  - Implement state for email input, loading state, success
10 state, and error state
11 - Create handlers for input change and form submission
```

New Session

Great! Now that we have the design approved, let's create the implementation plan with specific tasks:

Created `tasks.md`

Each task references specific requirements from the requirements document and focuses only on coding activities that can be performed within the development environment.

Do the tasks look good?

Move to the next phase

Ask a question or describe a task

Auto | Autopilot

# Практики волны спецификаций

## MCP: стандарт, который приняли все

Ноя 2024 → ноя 2025: **97M+ downloads**, **10 000+ серверов**
OpenAI, Google, Microsoft. → **Linux Foundation**

## Команды сжимаются

Google: **30-60 → 2-5**. Stripe CEO: «Minimum viable team collapsed»
**23%** перераспределяют бюджет headcount → AI tools

## Кризис джуниоров

Stanford: **-20%** занятость 22-25 лет. **44%**: падают фундаментальные навыки

## Prompt Engineer → ~~роль умирает~~

Indeed: 144/млн → **20-30/млн**. Навык: **68%** обучают. Роль: исчезает

# Центральный парадокс: adoption ≠ trust

**85%**

ИСПОЛЬЗУЮТ AI ЕЖЕДНЕВНО

**29%**

ДОВЕРЯЮТ РЕЗУЛЬТАТАМ

**66%**

«ПОЧТИ ПРАВИЛЬНО, НО НЕ СОВСЕМ»

Favorability: **77%** (2023) → **60%** (2025). «Willing but reluctant»

## METR RCT: 40 п.п. perception gap

Разработчики думали: **+20%** быстрее. Реальность: **-19%** медленнее

Sonar: toil = **23–25%** — **с AI и без**. Работа сместилась, не исчезла

> «AI magnifies the strengths of high-performing organizations and the dysfunctions of struggling ones.» — ***DORA 2025***

# Безопасность: цена скорости

## Package hallucination — «slopsquatting»

**19,7%** зависимостей в AI-коде — **несуществующие** пакеты (USENIX 2025)
**43%** повторяются → атакующие регистрируют с malicious code

## Уязвимости

**40%** GPT-кода с уязвимостями (Trend Micro). **73%** CWE (Georgetown)
CodeRabbit: vibe-coded PR = **1.7×** major issues, **2.74×** security vulns

## Доверие на цифрах

SO 2025: **3%** «высоко доверяют». **75%** ревьют каждый AI-сниппет
MCP CVE-2025-6514: **437 000+** скомпрометированных dev-окружений

# Пять волн — один каскад

```
Волна 1  TENSOR CORES   V100 → Transformer, RLHF → (4 года) → Copilot
Волна 2  МАСШТАБ         A100 → GPT-3, Scaling → ChatGPT → SO умирает
Волна 3  ЭФФЕКТИВНОСТЬ  H100 FP8, MI300X → MoE, KV → Cursor → Write Less
Волна 4  REASONING       B200, Groq → GRPO ★ → Claude Code, Codex → Agents
Волна 5  ВЕРИФИКАЦИЯ    NVFP4 → Astrogator → Kiro, Antigravity → Specs
```

Задержка: **4 года → 3 года → 2 года → 1 год → 6 мес**

Волны **перекрываются**: пятая начинается, когда третья ещё не закончилась

> *Hardware определяет всё.*
> *V100→Transformer. A100→GPT-3. H100→MoE. B200→Reasoning. NVFP4→Verification.*

# Что забрать с собой

**1.** Каждая волна начинается с **железа**. Следите за железом — оно предсказывает, куда пойдет наука, какие появятся продукты и практики на 1–3 года вперёд

**2.** GRPO — открытие уровня Transformer: **reasoning возникает** из чистого RL, без примеров

**3.** Три закона масштабирования **сходятся**: pre-training + post-training + inference. Оптимум — баланс по трём осям

**4.** AI = **усилитель способностей людей**, а не замена людей. 85% adoption, 29% trust. Работа **изменила характер**, не исчезла. METR: -19% при ощущении +20%

**5.** Будущее за **спецификациями + верификацией**. Domain-specific, не general. Spec-driven, не prompt-driven

# Цитаты, определяющие момент

> «Forget that the code even exists. I Accept All always.»
> — ***Karpathy***. *Collins Word of the Year 2025*

> «AI magnifies the strengths of high-performing organizations
> and the dysfunctions of struggling ones.»
> — ***Google DORA 2025***

> «A specification is a kind of (version controlled, human-readable) super prompt.»
> — ***Marc Brooker***, *Amazon*

> «Programming without these tools just feels primitive.»
> — ***Armando Solar-Lezama***, *MIT CSAIL*

# Уроки (горькие)

# Уроки: Волна Железа

- Качество результата зависит не только от твоего навыка, но и от инструментов
- Нет инструментов и железа - ничего не можешь сделать, и это нормально

# Уроки: Волна Масштаба - Публичные сервисы

- Claude Sonnet и Opus, тариф Max x20 - 200$ в месяц (API дороже)
- Grok Super Heavy - 300$ в месяц
- GigaChat Max - 9 750 ₽ за 15 миллионов токенов по API
- Любые средства для понижения стоимости!!!

# Уроки: Волна Масштаба - Локальное железо

Кринж

- Скорость инференса на CPU - 1-3 токена в секунду, это невыносимо
- RTX 4090/5090 стоят 200+ тысяч рублей, опасность расплавить питание
- Старые Tesla (K80, P100) - очень мало VRAM, не поддерживают FP16, а GGUF/llama.cpp работает на деквантизации в FP32, придется строить целый датацентр (минимум 40 карт)


NT-USB
$50,000

# Уроки: Волна Масштаба - Локальное железо

Дорого-богато

- Соберем всё на Exo + Mac!https://github.com/exo-explore/exo
  - MoE модели параллелятся хуже, чем densed — железа надо больше
- Максимум VRAM в MacMini
- 2-терабайтный массив из Mac Studio стоит около 10 миллионов рублей
- Аналогичный сетап на Nvidia/B200 стоит 40+ миллионов ₽, не найдёте в продаже
- Любые средства для понижения стоимости!!!
  - MoE, QLoRA, RAG...

# Главный способ экономить

Не стройте свой домашний датацентр

## Попросите работодателя купить учётки и железо

В AI-компаниях (Anthropic, OpenAI, GitVerse) у сотрудников нет лимитов на токены при использовании своих собственных инструментов

# Масштаб требует роста, рост требует скидок!

- Тарифы Claude и Grok - дотационные
- Cursor дешевый, но становится всё дороже
- GigaCode пока бесплатный

# Уроки: Волна Эффективности

- Ключевое: Выбор правильных инструментов
- Точный подбор модели
- Точный подбор железа
- Точный подбор методологии

# Уроки: Волна Reasoning

- Только Reasoning и Extended Thinking
- Точный подбор агентного тулинга
- Точный подбор модели
- Вайбкод для кастомных задач human-in-the-loop
- Background Agents для масштабных задач

# Уроки: Волна Спецификаций

- Spec-First: спецификации пишутся раньше кода
- Closed-Loop-Specs: спецификации обновляются автоматически
- Specs + Skills, оба вместе, а не что-то одно
- Использование очень конкретных инструментов (Kiro, Antigravity, GigaCode Agents, etc)

# Главный урок:

# Доверять нейросетям нельзя

(Но придётся)

И вот когда у нас есть на всё это ресурсы, инструменты и практики, когда мы умеем работать в ситуации неопределенности — 100% генерация кода ИИ возможна.

# 100% автоматическая генерация кода?

# Автоматически ≠ бесплатно

По факту, очень дорого

# Автоматически ≠ ничего не нужно знать

Огромный инструментарий, растянутый на 80 лет эволюции

И продолжает расти

# Автоматически ≠ без участия человека

Нужно писать спецификации, собирать железо, много всего

"Программистов" могут заменить,

но людям придется работать больше, а не меньше

# Новое лето искусственного интеллекта

ЧЕТЫРЁХСЛОЙНАЯ ВИЗУАЛИЗАЦИЯ — ФЕВРАЛЬ 2026

## *Hardware → Science → Products → Practices*

Как железо порождает открытия, открытия превращаются в инструменты, а инструменты меняют то, как разработчики и data scientists работают каждый день. Нажмите на узел — увидите связи и подробности.

HARDWARE • GPU / Accelerator | SCIENCE • Архитектура • Рассуждения • Инфраструктура • Верификация • Агенты | PRODUCTS • IDE / Copilot • Coding Agent • Платформа • Модель | PRACTICES • SWE • DS / ML • Обе аудитории

| | HARDWARE | SCIENCE & ENGINEERING | PRODUCTS | PRACTICES |
|---|---|---|---|---|
| **2017** Июн | **NVIDIA V100** • DEPLOYED<br>Первые Tensor Cores. **125 TFLOPS** для глубокого обучения. Матричные операции в 5–12× быстрее.<br>Без этого чипа Transformer остался бы академическим курьёзом. **Tensor Cores сделали attention реализуемым на практике.** | **Transformer — "Attention Is All You Need"** • DEPLOYED<br>Self-attention заменяет рекуррентность. **Параллельная обработка** всей последовательности. O(1) между любыми позициями.<br>**Фундамент всего.** Каждая модель и каждый продукт стоят на этой архитектуре.<br><br>**RLHF Foundations — Christiano et al.** • DEPLOYED<br>Обучение на основе человеческих предпочтений через reward models. **PPO** как алгоритм оптимизации.<br>Мост от «мощного движка» к «полезному инструменту». **Без RLHF модели не следовали бы инструкциям.** | | |
| **2020** Май | **NVIDIA A100 — 80 GB HBM2e** • DEPLOYED<br>Tensor Cores 3-го поколения. MIG. **2 ТБ/с** пропускной способности.<br>GPT-3 (175B) тренировался на кластерах A100. **Без 80 ГБ памяти модели такого масштаба были бы невозможны.** | **GPT-3 & Scaling Laws — Kaplan et al.** • DEPLOYED<br>175B параметров. Loss масштабируется степенным законом. **Эмерджентные способности** при масштабировании.<br>Scaling laws — «закон Мура для AI». **Предсказуемая отдача при увеличении модели в 10×.** | | **Transfer Learning → Fine-Tune** • DEPLOYED<br>BERT (2018) → GPT-3 (2020): **fine-tune pre-trained** вместо обучения с нуля.<br>Смерть кастомных word embeddings. **«Просто дообучи» — новый стандарт.** |
| **2021** Июн | | **Speculative Decoding** • DEPLOYED<br>Маленькая модель генерирует кандидатов, большая верифицирует параллельно. **Acceptance rate >85%.**<br>**2–3× ускорение inference** без потери качества. | **GitHub Copilot (Preview)** • DEPLOYED<br>Codex (GPT-3, дообученный на коде). **Первый массовый AI-ассистент для кода.**<br>Создал рынок AI-кодогенерации. **1M+ пользователей за год.** | **AI-автокомплит: первая волна** • DEPLOYED<br>Copilot preview: **27% кода** в enabled-файлах генерируется AI.<br>Новый воркфлоу: напиши комментарий → AI допишет. **Comment-driven development.** |
| **2022** Янв–Ноя | **NVIDIA H100 — FP8 & HBM3** • DEPLOYED<br>Tensor Cores 4-го поколения. **Нативный FP8.** 3,35 ТБ/с. 4× throughput vs A100.<br>FP8 — качественный скачок: обучение и inference в пониженной точности **без деградации.** | **Chain-of-Thought Prompting — Wei et al.** • DEPLOYED<br>«Let's think step by step.» **Промежуточные шаги рассуждения** кратно улучшают точность.<br>Простая идея с непропорциональным эффектом. **Фундамент reasoning.**<br><br>**InstructGPT — RLHF at Scale** • DEPLOYED<br>1,3B InstructGPT побеждает 175B GPT-3. **Post-training:** pre-train → SFT → RLHF.<br>**Маленькая модель с RLHF побеждает огромную без.** | **Copilot GA + ChatGPT** • DEPLOYED<br>Copilot выходит в GA (июнь). **ChatGPT запускается (ноябрь).**<br>Рынок кодогенерации **взрывается.** | **«Спроси AI» вместо Stack Overflow** • DEPLOYED<br>ChatGPT (ноя 2022): мгновенный ответ без осуждения. Вопросы на SO: **-11% за 2 месяца.**<br>SO → ChatGPT. **PNAS Nexus: -25% снижение** активности SO. |
| **2023** | | | | |

DEPLOYED

# Transformer — "Attention Is All You Need"

НАУЧНАЯ СУТЬ

Self-attention заменяет рекуррентность. **Параллельная обработка** всей последовательности. O(1) между любыми позициями.

ЦЕННОСТЬ ДЛЯ ИНЖЕНЕРА

**Фундамент всего.** Каждая модель и каждый продукт стоят на этой архитектуре.

ССЫЛКИ

**Attention Is All You Need ↗**
Оригинальная статья Vaswani et al. (NeurIPS 2017). Архитектура, изменившая всё машинное обучение.

**The Illustrated Transformer ↗**
Jay Alammar's визуальное объяснение — лучшее введение в механизм self-attention.

**The Annotated Transformer ↗**
Harvard NLP: построчная реализация Transformer на PyTorch с объяснениями. Для тех, кто хочет понять код.

— ОСНОВАН НА

ПОЗВОЛИЛ NVIDIA V100 →

— РАЗВИТИЕ

ЭВОЛЮЦИЯ GPT-3 & Scaling Laws — Kaplan et al. →

ПОЗВОЛИЛ GitHub Copilot (Preview) →

# Developer-first платформа с AI, доступная в облаке

# GITVERSE

# Спасибо. Вопросы?

GitVerse с ИИ внутри: https://gitverse.ru

Эти слайды:
https://oleg.guru/talks/100-percent-generation

Новое лето искусственного интеллекта:
https://oleg.guru/ru/timeline

80 лет борьбы чата с файлами:
https://oleg.guru/ru/chat-vs-file

Промт для исследователей:
https://gist.github.com/olegchir/c1104526b000be33763b6369d20f0ebc

Остальные ссылки (Telegram, Twitter...):
https://oleg.guru