History of Artificial Intelligence

From Expert Systems to the Age of Generative Models (GenAI) – Research Report for software engineers entering the market in 2025–2026.

Why is this important?

AI as a requirement, not an option

The new role of the engineer

Pure "coding" is becoming automated. The modern programmer is a systems architect and business logic verifier—the so-called AI-Assisted Developer.

Adoption Rate (2025)

- GitHub Copilot: 20 million users, 400% growth year-on-
- year ~90% of Fortune 100 companies use AI assistants
- 46% of new code generated by AI In Java, this figure is
- 61%, in Python 55%

The Impact of AI on Productivity

52%

Confirmed growth

Developers Confirm Positive Impact of AI on Performance (Stack Overflow 2025)

55%

GenAI programmers complete tasks faster in controlled studies

75%

Shorter code review

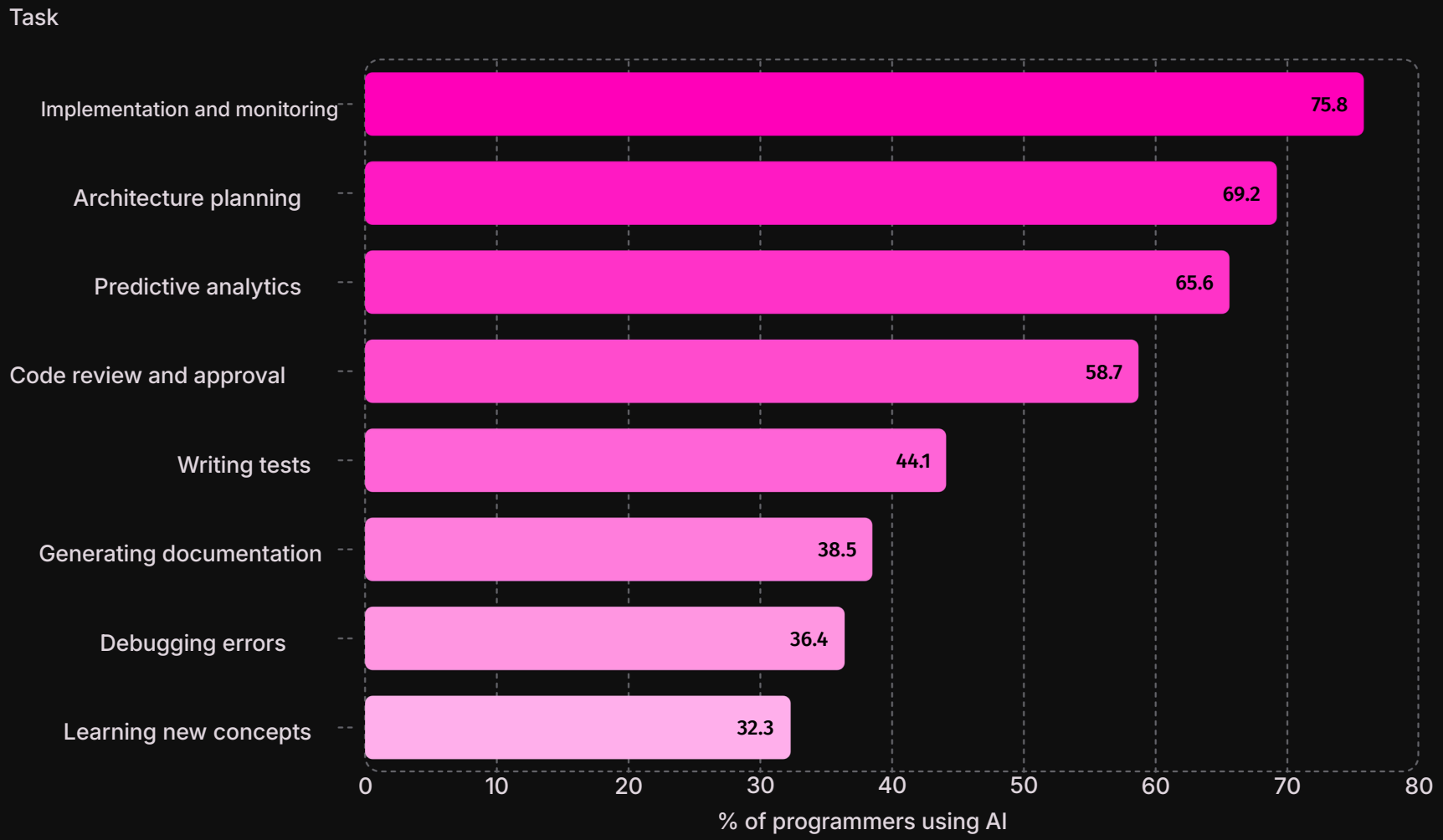
Code review time drops from 9.6 to 2.4 days

817K

Positions in Poland

highly susceptible to interaction with generative systems (NASK report)

AI Usage by Task Category (2025)



Data from the Stack Overflow 2025 report. The highest adoption rate is observed in operational and architectural tasks – AI is becoming the standard throughout the software development lifecycle.

CHAPTER 1

Prehistory and the birth of AI

1

1950 — AlanTuring

Essay Computing Machinery and Intelligence. Proposal "Games in Imitation" - The Turing Test as a benchmark for assessing machines.

2

1955–56 — Logic Theorist

Newell and Simon create a program that proves 38 theorems from *Principia Mathematica. One proof—shorter than the author's.*

3

1956 — Dartmouth

McCarthy, Minsky, Shannon. The Official Birth of the Discipline of Artificial Intelligence. Great Optimism—Full Intelligence in a few weeks?

4

1964–67 — ELIZA

Weizenbaum (MIT) creates the first chatbot. Matching Regex Patterns Trigger the "ELIZA Effect" — Users they confide in the machine.

ELIZA and the Turing Test—Two Lessons in Humility

Test Turin (1950)

A judge talks to a machine and a human through a terminal. If the judge can't tell the difference, the machine is "intelligent." Modern LLMs pass this test easily, lacking consciousness—imitation is algorithmically simpler than general rationality.

ELIZA (1966)

Architecture based 100% on **pattern matching. Pattern I** feel X → odpowiedź Why do you feel X?. Inability to understand words. Yet, users confided intimate details of their lives to the machine—the ELIZA Effect.

First Winter of AI

The combinatorial explosion stopped everything

As algorithms moved beyond the closed logical environments, they encountered a combinatorial explosion—the number of decision states grew faster. than exponentially, and the 1970s hardware had minimal RAM. The Lighthill Report (1973) brutally summarized the impossibility of scaling outside the laboratory. Sponsors cut off funding—the First AI Winter had arrived.

CHAPTER 2

The era of expert systems

Knowledge Base

Thousands of facts and rules IF A and B → THEN C, hand-coded by programmers based on interviews with experts.

Inference Engine

Dynamic processing module matching rule chains to user input, independent of the knowledge base - you can update rules without breaking the engine logic.

The 1980s saw the abandonment of the "thinking machine" dream in favor of specialized commercial tools. Expert systems had a single purpose: to emulate an expert in a very narrow field.

DENDRAL and MYCIN – Triumphs and Barriers

DENDRAL (Stanford, 1960s)

Analyzing mass spectrometry results, he deduced the structure of organic molecules with incredible accuracy—results many times greater than those of novice researchers in this narrow niche.

MYCIN (Stanford, 1970s)

Diagnosis of bacterial blood infections and dose selection antibiotics. In blind tests, more accurate than specialists. Rule: *"IF Gram stain negative AND bacillus AND fever → intestinal infection (0.8 confidence)".*

Why wasn't MYCIN distributed to hospitals?

The problem wasn't the code—it was the legal and ethical issues. Who was responsible for a patient's death due to an algorithm? Doctors didn't trust computers enough to entrust them with human lives.

Second Winter AI (late 90.)

Rule-Based Systems Hit a Wall: Inability to Learn, a Problem brittleness, huge costs of maintaining the knowledge base. Hand coding the complexity of the world has proven to be architectural absurdity.

CHAPTER 3

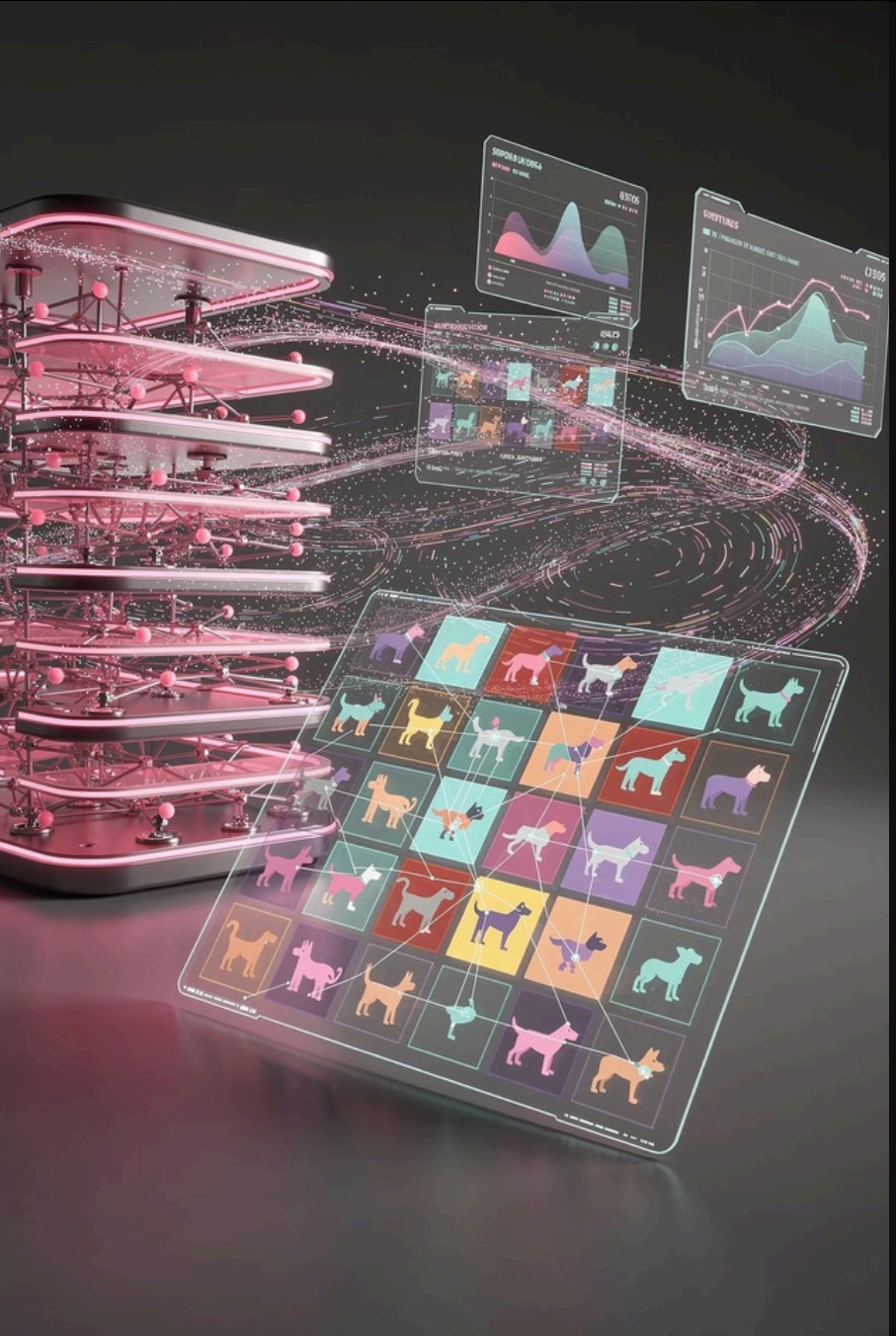
The Big Data Revolution and Machine Learning

Reverse programming logic

In the traditional paradigm: rules + data → result. In machine learning: data + correct **responses → rules discovered by the model. Moving from predetermined commands to adaptive probability matrices.**

Two catalytic converters

- **Big Data: Internet, social media, IoT – unimaginable amounts of unstructured data** as training fuel
- **GPU: Thousands of Simple Cores Working in Parallel - Neural Network Training** accelerated a thousandfold



AlexNet 2012 - Deep Learning Awakens

Bankruptcy ImageNet

September 30, 2012. Classification of hundreds

thousands of photos across 1,000 categories. AlexNet (Hinton, Toronto) crushed the competition with a 9.8 percentage point lower error.
from second place.

An engineering masterpiece

Training divided into two cards **NVIDIA GTX 580. ReLU activation solved the "vanishing gradient" problem — enabling the training of deep, multi-layer networks.**

Layer hierarchy

First layer: boundary lines. Next: textures and shapes. Last: final prediction (e.g., "Siberian Husky"). The boom in multilayer neural networks.

AlphaGo vs Lee Sedol — The Rubicon Crossed (2016)

Why was Go the "Holy Grail"?

The number of move combinations exceeds the number of atoms in the universe. Brute-force methods (like Deep Blue in chess) were useless. DeepMind combined Monte Carlo + deep neural networks + reinforcement learning. The result: 4–1 for AlphaGo.

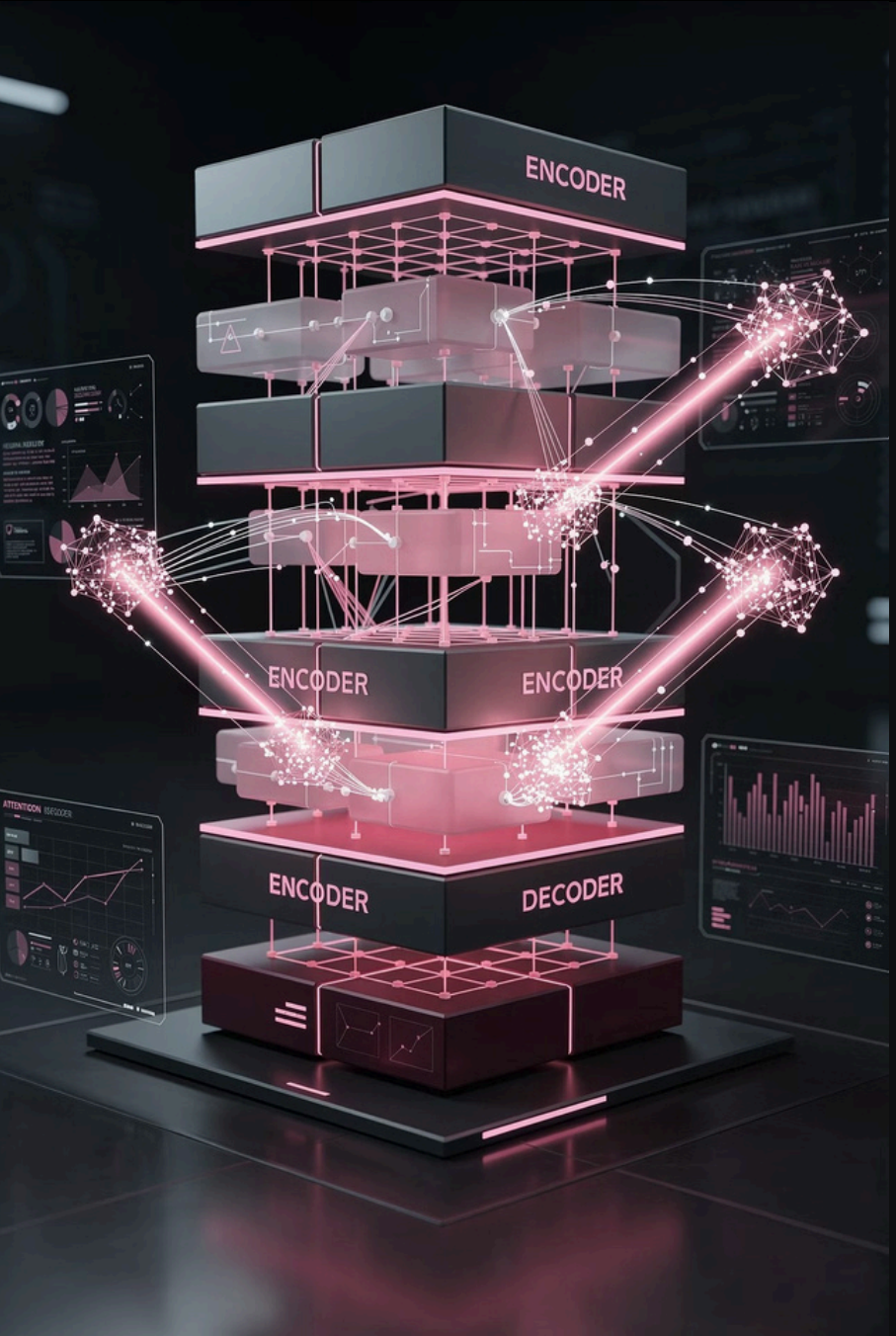
Two iconic moves

Move 37 (game 2): AlphaGo played illogically—the commentators considered it a mistake. After several dozen moves, it turned out the machine had designed long-term territorial control. The machine's first "creative intuition."

Move 78 (game 4): Lee Sedol made a desperate, statistically rare

move

move — "God's Touch." AlphaGo got completely lost and lost. Proof of the flexibility of the unstructured human mind.



CHAPTER 4

Transformer Revolution — "Attention Is All You Need" (2017)

Until 2017, recurrent RNN/LSTM networks dominated – they read text sequentially, losing context from the beginning of long documents. The GoogleBrain article changed that completely.

Query (Q)

What a given token is trying to find relative to others in context.

Key (K)

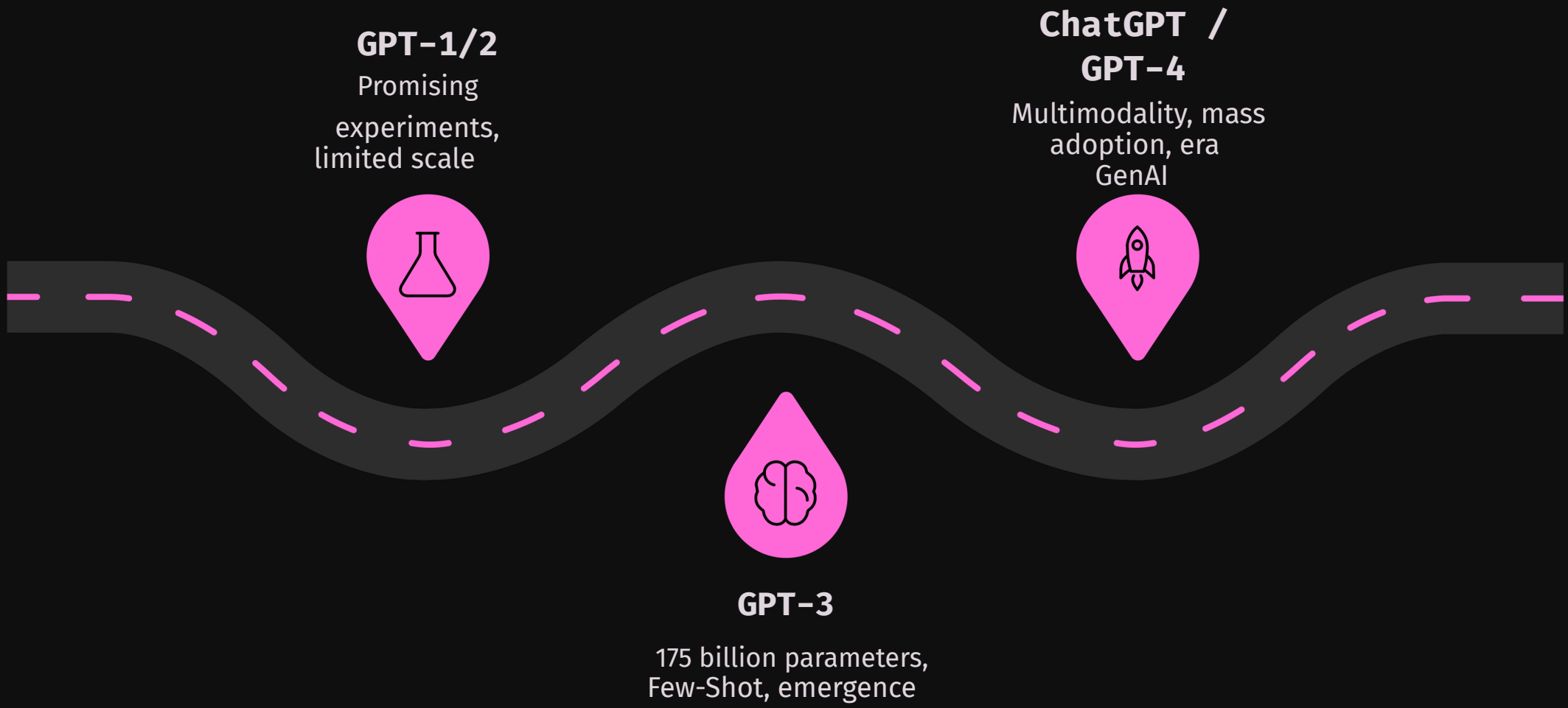
Features that tell the rest of the system, "here's exactly what this item contains."

Value (V)

The true weight vector of the concept passed in the final process.

Thanks to QKV, the sentence "The zipper is stuck on my jeans" is 100% indifferent to the double meaning of the word "zipper"—the system assigned a high weight to the word "jeans." The article reached ~173,000 views by 2025. **citations.**

From GPT-3 to ChatGPT – the birth of the LLM

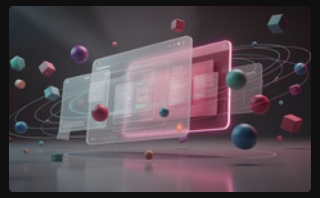


OpenAI demonstrated between 2018 and 2020 that model intelligence grows exponentially with the number of parameters. GPT-3 (175 billion parameters) revealed the phenomenon of emergent intelligence—few-shot learning: the machine deduced abstract phenomena from a few examples without expensive fine-tuning. GPT-4 added multimodality, and ChatGPT sparked the mainstream GenAI explosion in 2022.

AI Model Landscape (2025–2026)

Claude (Anthropic)

Versions 3.5 and 4.5 of Sonnet. Dominates programming logic benchmarks, especially front-end benchmarks with modern frameworks. Maintains precise analytical direction.



Gemini (Google)

Powerful Context Window - Gemini 1.5 Pro consumes thousands of pages or vast code trees at once, ensuring architectural consistency without losing variables deep in subdirectories.

Mistral / LLaMA (Meta)

The Free License Revolution and Mixture-of-**Experts (MoE) — activates only a dedicated subset of parameters (e.g., 41 billion out of 675 billion). Drastically reduces API costs while maintaining high accuracy.**

Diffusion Models - Image Generation

How do Diffusion Models work?

Forward diffusion: imaged pixel by pixel to "random noise soup" (Gaussian noise).

Reverse diffusion (Denoising): U-Net learns to perfectly predict deletion
noise, conditioned by a text vector (Cross-Attention).
The result: conjuring a hyperrealistic portrait from noise simply by a text request—

e.g., "astronaut on coral." Commercially available: DALL-E, Midjourney.

CHAPTER 5

AI-Assisted Developer Tools (2025–2026)

GitHub Copilot (Microsoft)

Enterprise-standard, integrated with repositories. Built into Visual Studio, it anticipates large boilerplate code blocks and solves repetitive problems without manual intervention. 20 million users.

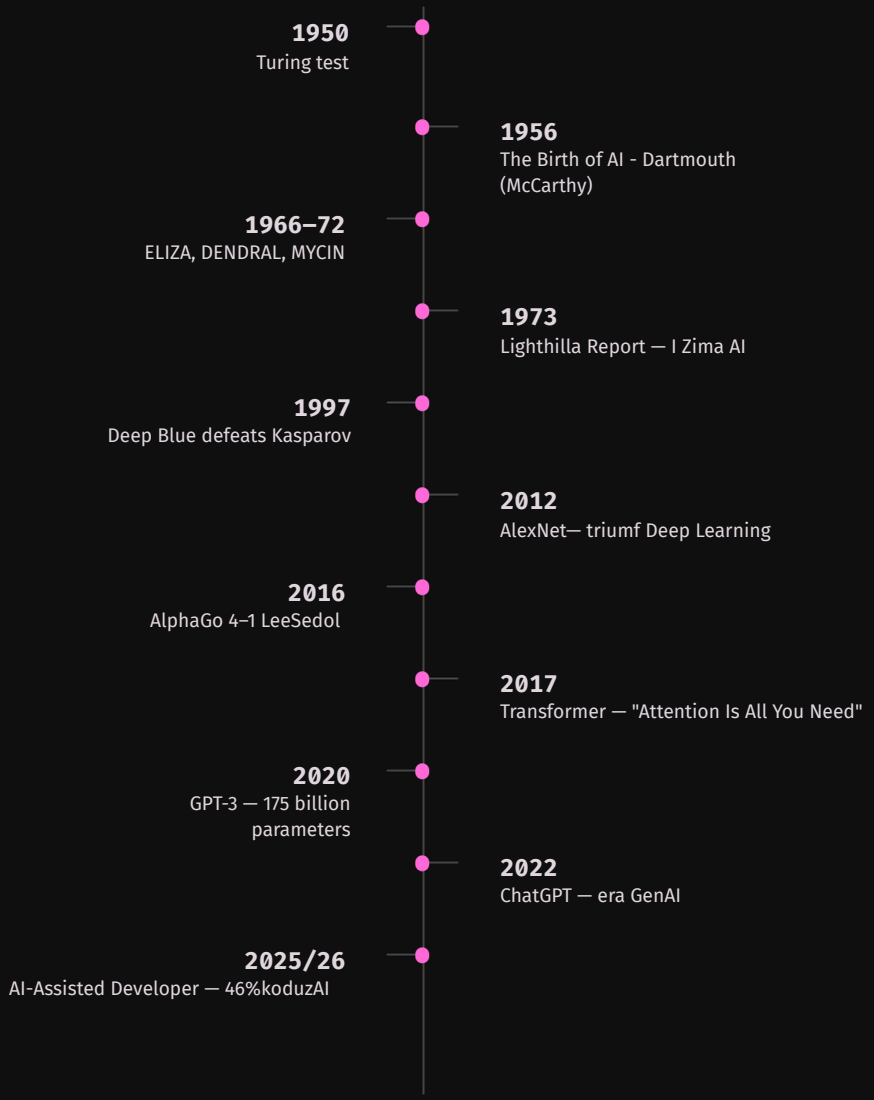
AI cursor

The king of advanced editors, based on VS Code. A single CMD+K command creates a coherent service with unit tests. Dynamic changes can be made simultaneously to dozens of files.

Windsurf (Codeium)

A favorite of developers with open commercial plans. Exceptional in projects requiring iron-clad contextual memory—it keeps the operator informed step by step through the flow of activities, "never losing the thread" of engineering.

AI Milestones Timeline



PRACTICAL EXERCISES

Three tasks for a future engineer

1 Naive rule system in Jupyter

Build an ELIZA-inspired chatbot using only dictionaries and the re (regex) library. The input "I feel scared about my job" → dynamic string f-formatting. Docblock: why regex has nothing to do with deep vector semantic binding Deep Learning.

2 Contextual Engineering – benchmarking LLM

Paste the buggy asyncio code with the deadlock (silent-bug). Design a step-by-step prompt explaining the assumptions. Compare two models (e.g., OpenAI vs. Anthropic) in a Markdown table—which one hallucinated, which one maintained the conversational vector.

3 Technical Essay (~300 words, .md file)

Which paradigm shift (hard rules → ML → Transformer → agent-based IDE) has the greatest disruptive impact on your engineering environment? Maximum points: appeal to an "empty knowledge base" vs. the simultaneity of the QKV mechanism. Critical analysis of AI model distortions required.

Key takeaways

Understand history

Every architectural limitation of today's LLMs has its roots in history—from the fragility of rule-based systems to the vanishing gradient problem.

Be an architect

Not a "code whiz." Define problem, instruct models, audit the security and integration of the generated code.

Maintain a critical horizon

The weightless token model will serve up a flawed architecture without a doubt. Your value lies in your ability to detect the machine's hallucinations.

For a company or developer to ignore AI in 2025 is tantamount to being forced out of business—the productivity dissonance is disproportionate. Your advantage is understanding this timeline.