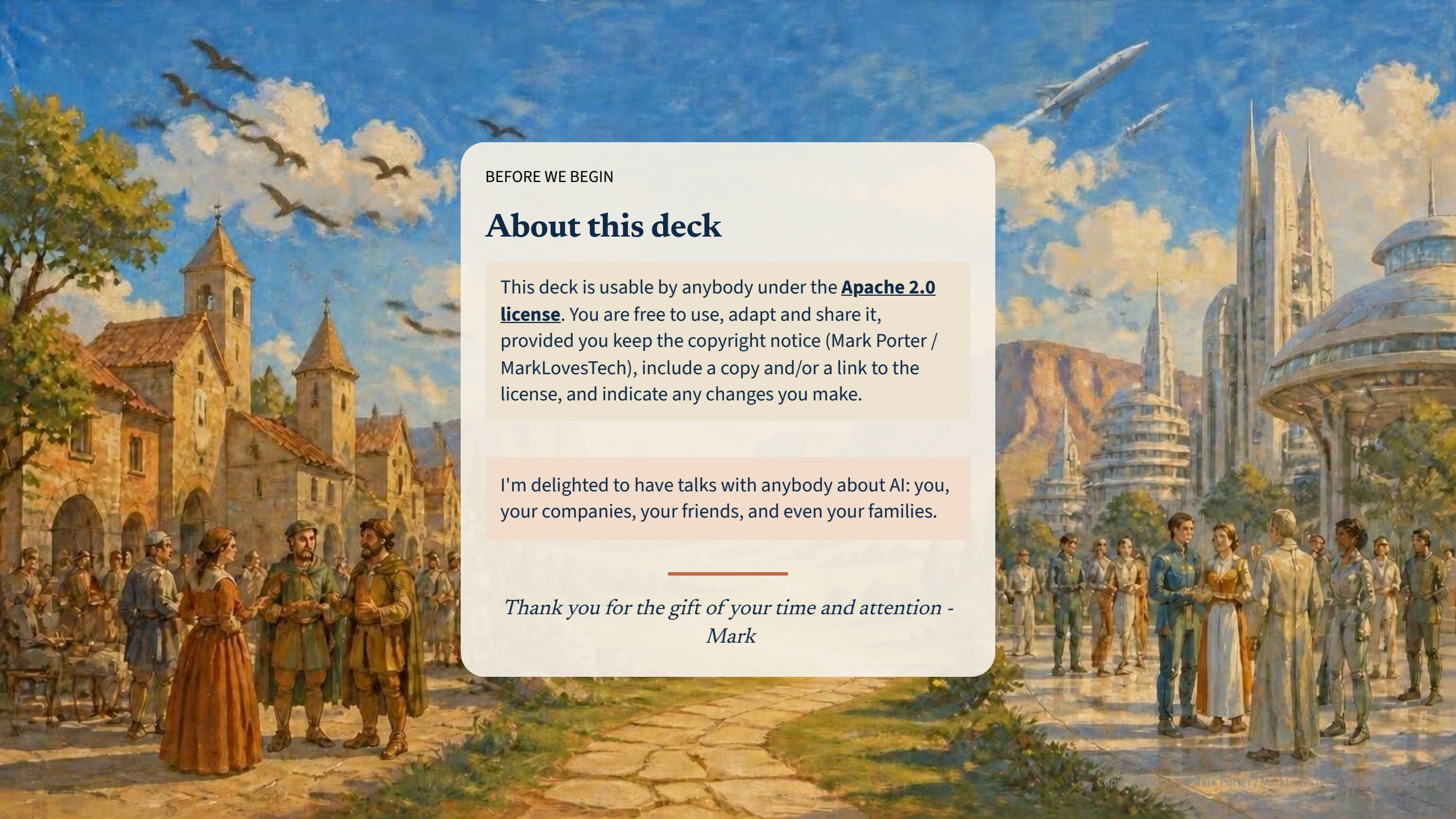


BEFORE WE BEGIN

About this deck

This deck is usable by anybody under the **Apache 2.0 license**. You are free to use, adapt and share it, provided you keep the copyright notice (Mark Porter / MarkLovesTech), include a copy and/or a link to the license, and indicate any changes you make.

I'm delighted to have talks with anybody about AI: you, your companies, your friends, and even your families.

Thank you for the gift of your time and attention - Mark

Copyright (C) 2024 Mark Porter / MarkLovesTech. Apache 2.0 License

HEALDSBURG · 2026

AI: Where It's Been, Where It's Going, and What You Can and Should Do Now

What AI is really doing to software and the companies that build it

Mark Porter, Global Field CTO, Cognition. Former CTO and GM: AWS, MongoDB, dbt Labs, Grab.

Wednesday, October 21, 2026

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Who am I?

Mark Porter

Over 40 years of data, systems, teams and cultures

Developer / Nerd

Designer

Executive

DBA

Founder

General Manager

CTO

Board Advisor

Board Member

Field CTO at Cognition

'74-'88 Early years Started at 10, professionally at 16; Apple Pascal programmer, Caltech before there was a CS degree, NASA/JPL, US Dept. of Education

'88-'00 Oracle 11th database engineer; Media Server founder, MPEG video patents

'01-'09 CastleAV founder, home-automation business

'11-'13 Amplify (NewsCorp) CTO, Insight division

'13-'18 AWS General Manager, RDS, Aurora, Database Migration Service; Aurora and RDS: $50M to $7B, 2nd largest AWS service.

'18-'20 Grab CTO, Mobility and Core Technology. The "Uber" of Southeast Asia

'20-'23 MongoDB board member, then CTO

'24-'26 dbt Labs CTO

'26-now Cognition Field CTO

Boards and advisors: Super (now), GitLab, MongoDB, Splyt. MariaDB, >commit.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

What I owe you from today

📜 History of AI

How we got here, 1890 to today

🎉 Some fun nuggets

Be the life of your AI party

🔥 My opinions

Where the AI space stands now

🚀 Your own AI journey

Data, questions and opinions to drive it.

FAR MORE IMPORTANT

Become the AI GOAT in your conversations

Drive piercing and rational AI conversations with your portfolio companies

...and with your friends and family

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

A QUICK PUBLIC SERVICE ANNOUNCEMENT

Put the phone and the notepad down.

Yes, there is a lot of text.

Yes, that is on purpose.

No, you do not need to photograph it.

Grab your wine / mimosa / beer and relax :-)

(I'll distribute the slides to you)

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Software engineering and AI are inextricably linked

● AI ● Software

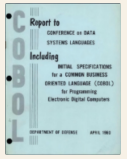
● 1890 Punch-card census tabulation

● 1949 Assembly language

● 1956 Dartmouth names AI

● 1958 Lisp, the AI language

● 1959 COBOL

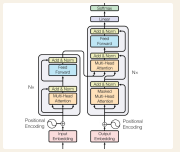


● 1966 ELIZA, first chatbot

● 1995 Java

● 2012 AlexNet learns to see

● 2017 The Transformer paper



● 2022 ChatGPT arrives

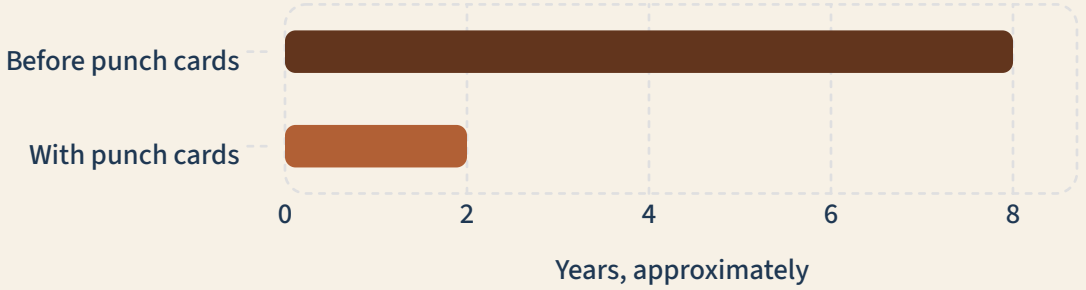
● 2030 ?

Agents as teammates?

Copyright (C) 2025 Mark Porter / MarkLovesTech. Apache 2.0 License

1890: the first time an electric machine helped IT move faster

Census processing time (years)



1888 Contest	Hollerith	Rival A	Rival B
Prep for tabulation (hours)	5.5	44.5	55.5

1. Punch one card per person
2. Read holes via electric circuits
3. Count each trait on a dial
4. Eliminate human counters completely
5. Fun fact, the machine used open pools of mercury as conductors

Source: US Census Bureau

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

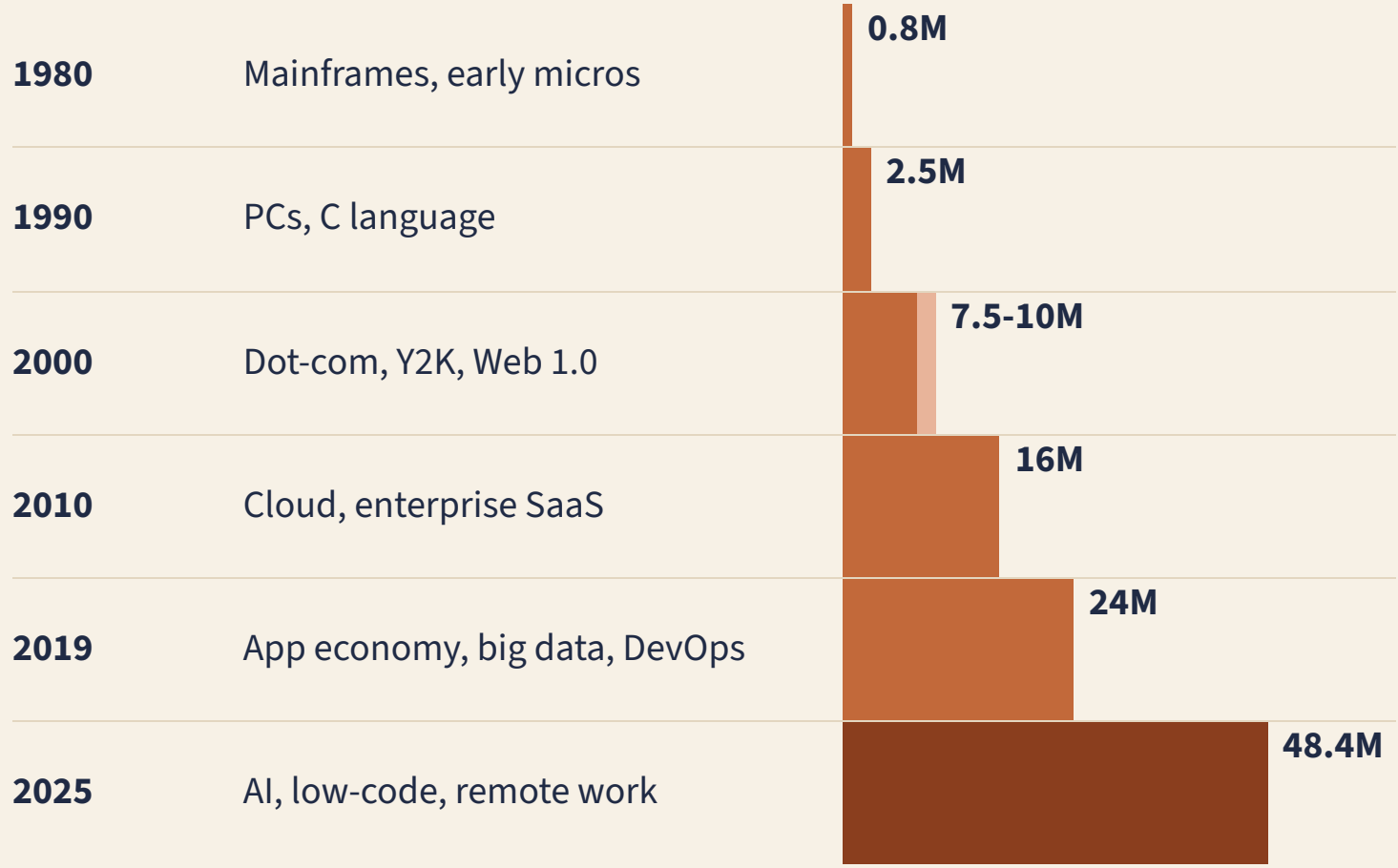
Every time computers get better, the end of human contribution is predicted

Era	What changed	The prediction
1960s · COBOL	**Assembly → COBOL, FORTRAN**	~~This is the end of programmers.~~
1980s · 4GLs	**COBOL → 4GLs (SQL, report writers), C, C++, Pascal**	~~This is the end of programmers.~~
2000s · Low-code	**In-house → offshore, code → low-code**	~~This is the end of programmers.~~
2022 · Copilots	**Editor → AI copilot (GitHub Copilot)**	~~This is the end of programmers.~~
2025 · Agents	**Copilot → autonomous agents** AI works independently, longer	~~This is the end of programmers.~~
2026 →	**Agents → agent teams** Agents do all manual work; humans bring taste, judgment	**?**

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

The appetite for software is insatiable

Estimated active software developers worldwide, millions

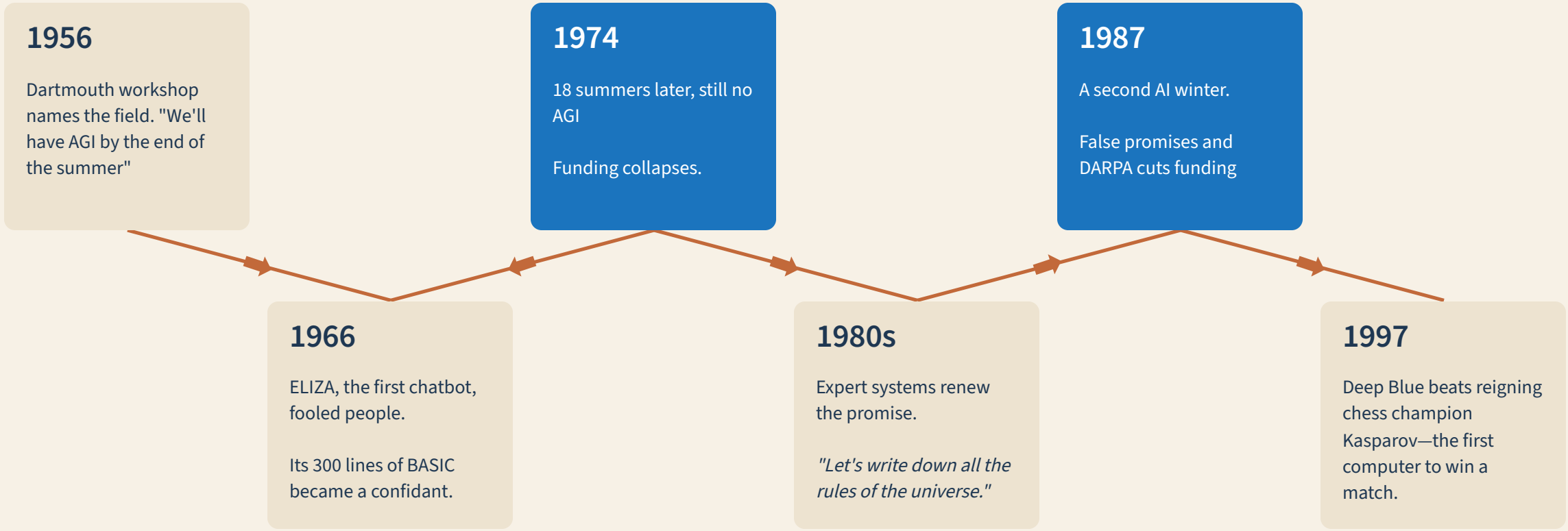


Up about 60-fold since 1980, with more programmers after every wave of automation. Source: SlashData; earlier decades are estimates.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

AI's first 40 years: big promises, two winters

1950 TO 1995 · PROGRESS AND FUNDING MOVED AT DIFFERENT SPEEDS



The colossal mistakes and hubris of these 40 years are only matched by... the mistakes and hubris of today.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Ignoring the shade thrown at it, in the 1990s, machine learning and neural networks ran your business

1997: Deep Blue beats Kasparov

Spam filters

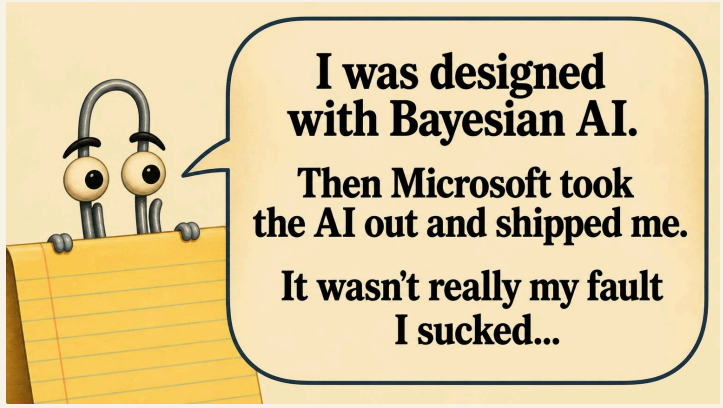
Credit and fraud

Search ranking

Recommendations

Demand forecasting

It worked so well that nobody called it AI.



THE ENGINE UNDERNEATH

Neural networks

Minsky called them a dead end in 1969 (for single-layer networks).

Hinton won the 2024 Nobel Prize in Physics for them.

Today's AI, LLMs included, runs on them.

THREE JOBS THAT DID NOT EXIST

Data scientists: find patterns

Data engineers: build pipelines

AI engineers: ship models

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

LLMs get all the attention - this is wrong for two reasons

Vision

Recognizes images

Games and planning

Plans many moves ahead (Deep Blue)

Science

Predicts protein folding, solving math problems

Forecasting, optimization

Demand, fraud, routing

Robotics and sensors

Self-driving cars, factory robots

World models

Predicts how the physical world behaves

Language

Chatbots like ChatGPT, (LLMs)

Sources: Fedorenko lab (MIT); Huth, Gallant et al. (UC Berkeley, 2016)

LANGUAGE IN THE HUMAN BRAIN

~5%

of the brain's volume is the language network (Fedorenko lab, MIT; fMRI of 772 people)

1/3+

of the cerebral cortex is involved in word meaning (Huth et al., UC Berkeley, 2016)

Language is not thinking

People with aphasia lose language but can still reason and plan.

Don't expect LLMs to do our thinking for us.

Source: Andrej Karpathy, deep dive into how LLMs work, youtube.com/andrejkarpathy

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Let's be clear - attention was really all we needed

4

2025 **Agents**

DeepSeek-R1 (open), Claude Code, GPT-5

WORKS FOR **Hours**

3

2024 **Reasoning**

OpenAI o1, Claude 3, Llama 3

WORKS FOR **Tens of minutes**

2

2023 **Better and open**

GPT-4, Claude, Llama 2 (open)

WORKS FOR **Minutes**

1

Nov 2022 **Chat**

ChatGPT launches

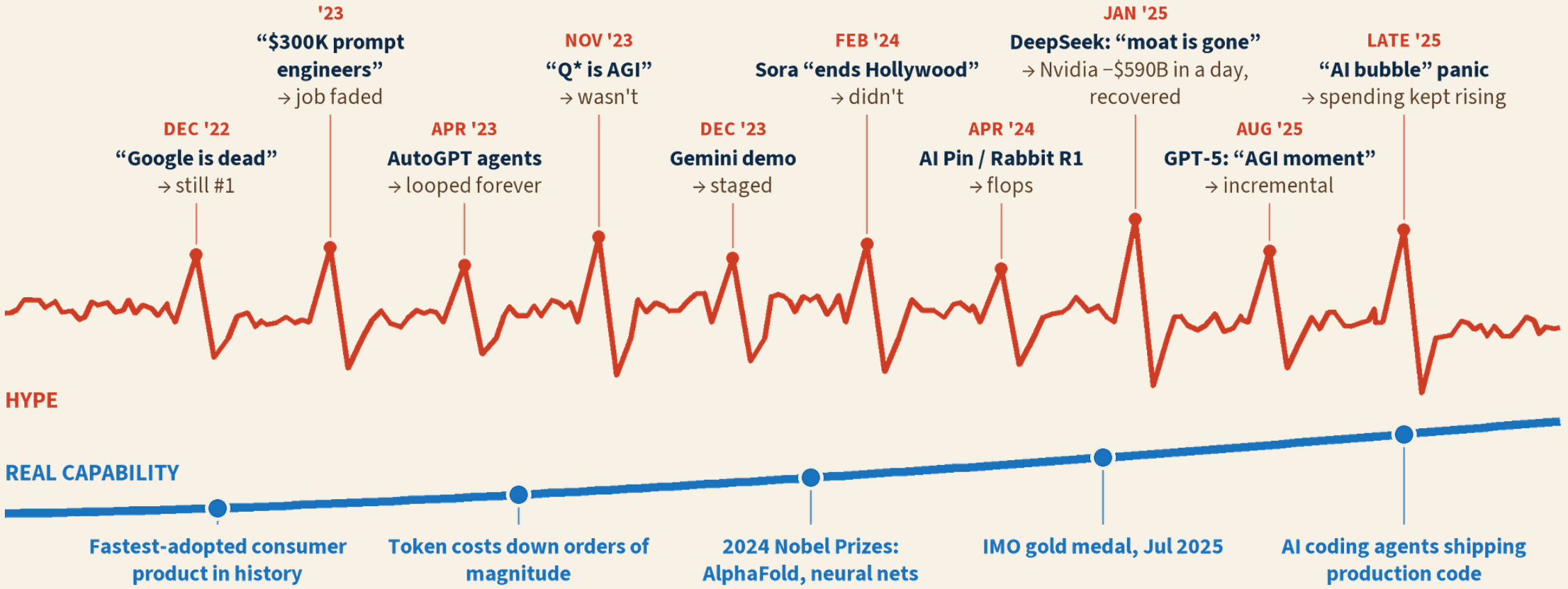
WORKS FOR **Seconds**

Attention Is All You Need (Vaswani et al., 2017)

In three years: from seconds of thinking to hours of work.

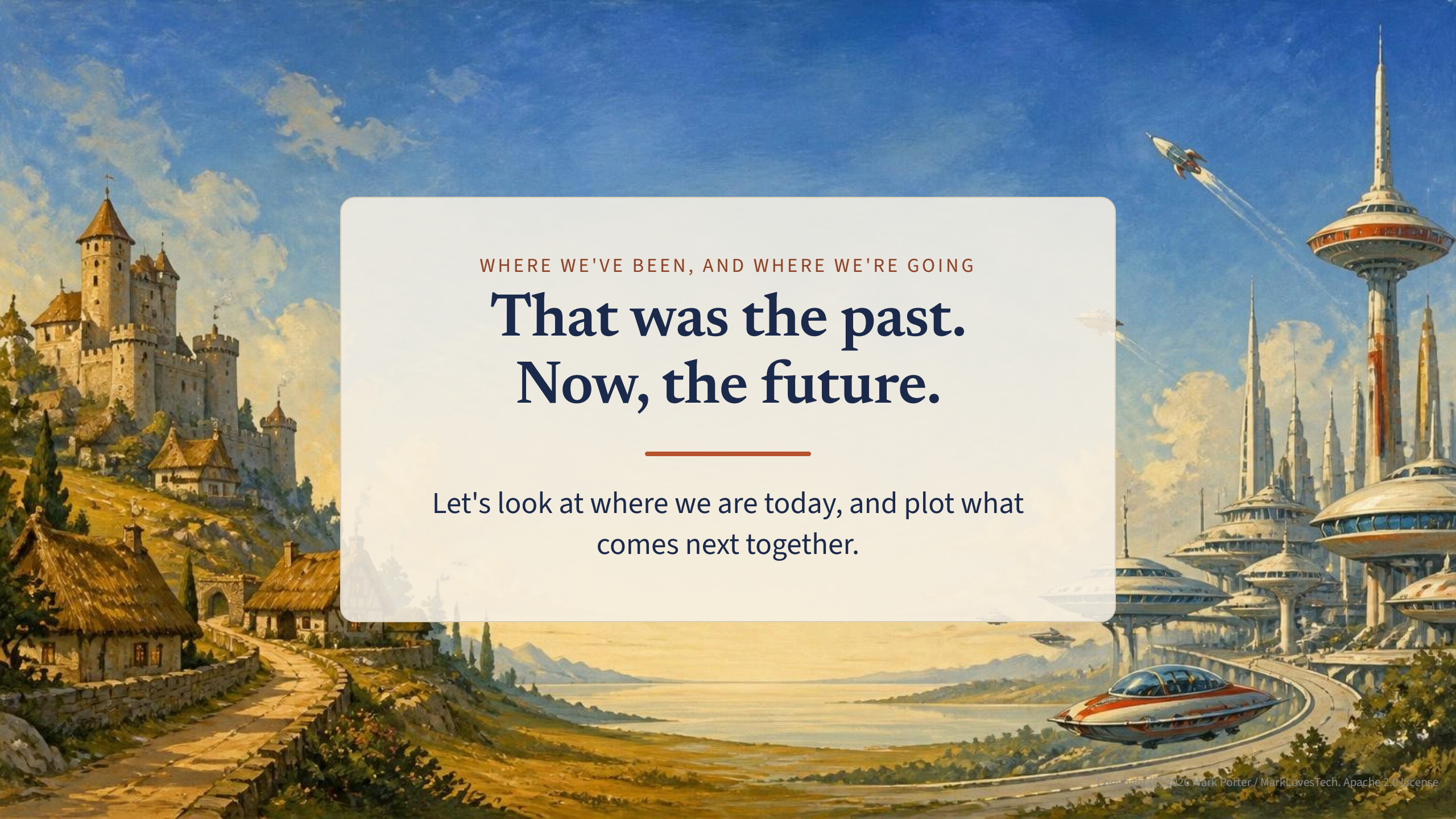
The Hype Was Wrong. The Technology Is Real.

LLMs since ChatGPT, Nov 2022 – 2026



Ignore the spikes. Watch the slope.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License



Copyright © 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Opinion #1: "TokenMaxxing" was and is a bad idea

Tokens burned measure fuel, not distance.

Leaderboards and consumption mandates get gamed.

Spend climbs without a link to value, leaving code nobody understands: comprehension debt.

Reasoning → agents → parallel agents: each step multiplies token use about 10x.

Measure outcomes per dollar.

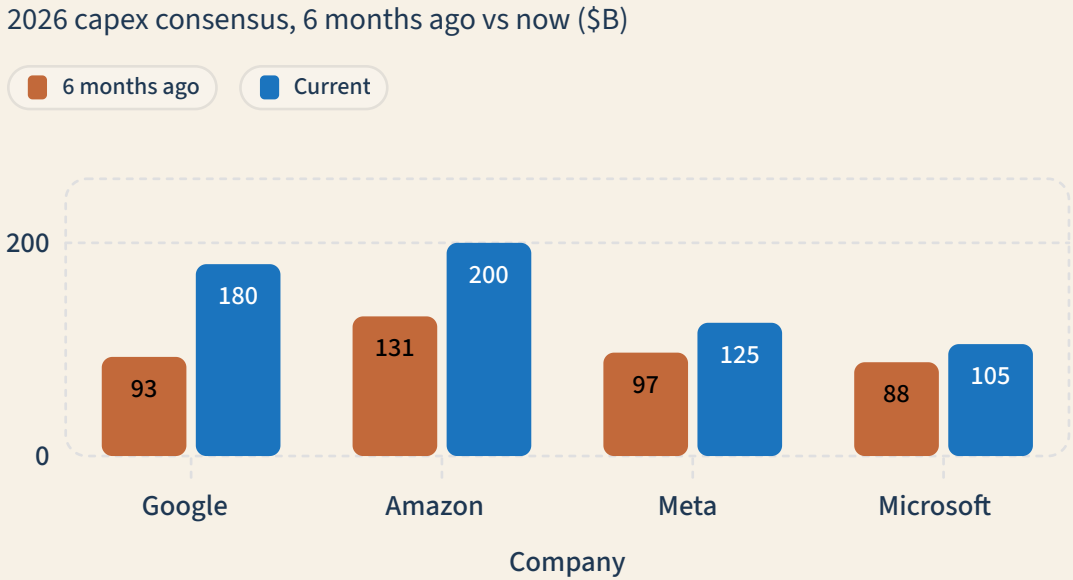
Shipped, reviewed, secure changes.

MAX TOKENS ≠ useful distance

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Opinion 2: Many DataCenter deals are at risk in one way or another

Any bank can tell you how this ends.



Debt and special-purpose vehicles fund the gap. Power, datacenters and fabs add physical fragility.

The loop: chipmakers → cloud providers → model labs → lenders and SPVs → neoclouds → customer revenue.

Moving bottlenecks: power (2024) → datacenter space (2025) → fabs (2026) → EUV (2027).

Our prediction: a correction in the infrastructure financing layer is likely.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Opinion 3: Local compute may start dominating the world

Most AI work will run on the devices we already carry. Datacenters stay for the hardest problems.

Simpler models ← → more complex models

Simple neural nets

Good for: keyword spotting, sensors, simple classification

Runs on: microcontroller (~20 KB model)

Decision models

Good for: routing, extraction, rewriting, short summaries, small offline assistants

Runs on: phone (1–4B parameters, ~3–4 GB)

Larger models

Good for: richer assistants, writing and analysis, coding help

Runs on: laptop (Gemma 4 12B, ~8 GB)

Frontier-scale

Good for: designing materials and drugs, understanding cancer

Runs on: datacenter (Qwen3 235B ~142 GB; DeepSeek V4.1-Flash ~510 GB)

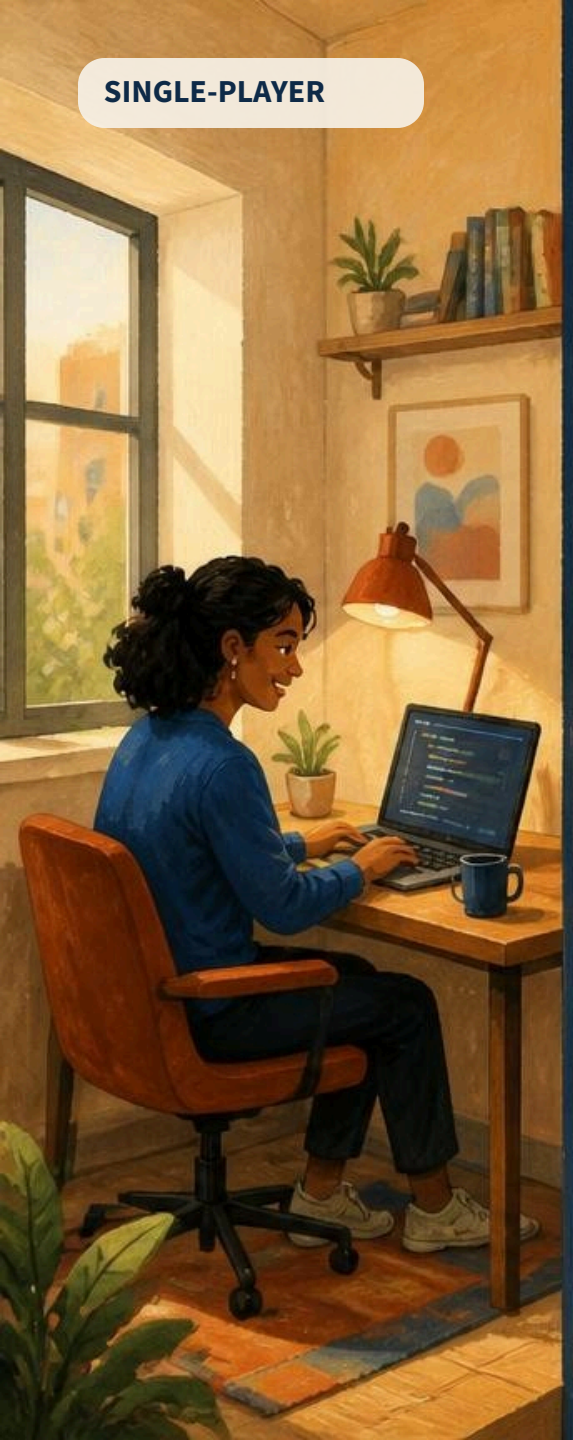
Our view: the first three segments cover the vast majority of what people want from AI, and they can run on phones and laptops. Only the far right needs datacenter-scale compute.

Projection, not fact: 1–4 TB phones could make on-device AI the default within a few years.

And that keeps all my data local. What happens on my phone stays on my phone...

Sources: csuite.so/blog/local-ai-on-every-size-of-machine · slash-digital.io/en/insights/small-models-on-device · adroit.io/field-notes/on-device-small-language-models-2026 · basecompute.co/stateoflocalai

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License



OPINION 3

Improve both single-player and multiplayer

Single-player

One developer, one laptop, glorious code.

Multiplayer

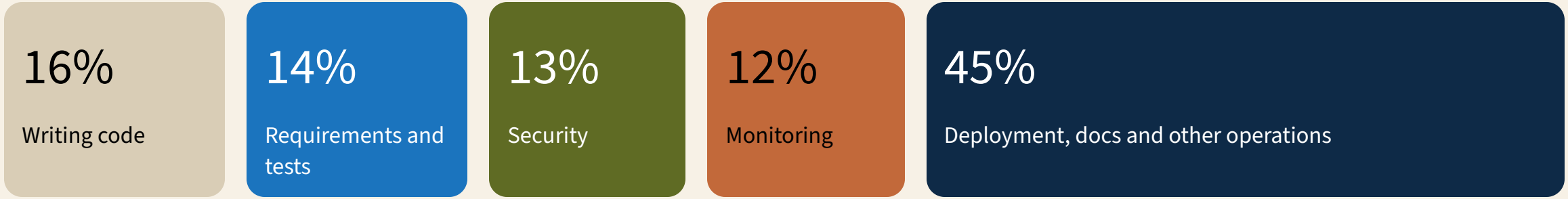
Teams that build together, and teams that watch the dashboards.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Coding is only ~16% of the job

Where software engineers spend their time

Where most AI tools focus today



84%: the real opportunity for AI

The math has a ceiling

Even if AI made coding instant, teams would get at most ~19% faster (Amdahl's law).

Another view

Microsoft's Time Warp study puts coding at just 11% of the actual workweek.

Speeding up the keyboard can only move the needle so far.

Sources: IDC, How Do Software Developers Spend Their Time? (2025; remaining 45% is other operational work); Microsoft Time Warp (2025).

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Make single-player better: aim AI at the other 84%

Optimize the whole lifecycle, not just the keyboard

Savings get cancelled

68% of developers save 10+ hours a week with AI, yet 50% lose 10+ hours a week to friction like finding information. (Atlassian State of DevEx 2025)

AI amplifies the system

AI raises delivery throughput but still hurts stability unless testing, review and operations improve alongside it. (DORA 2025)

Aim at the whole lifecycle

Requirements, testing, security, monitoring, deployment and finding information: where most of the time goes.

Requirements → Review and testing → Security → Operations → Knowledge discovery

Optimize the whole software lifecycle, not just the keyboard.

Sources: Atlassian State of DevEx 2025; Google DORA State of AI-Assisted Software Development 2025.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

WHERE ENGINEERING TEAMS ACTUALLY SPEND THEIR CAPACITY

Software is a team sport. AI has to play every position.

PLAYMAKER
Design and plan
Maps the codebase and architecture before work starts

Where team AI plays

SCORER
Write code
Takes the ticket and executes, in parallel

Where coding assistants play

DEFENSE
Review and security
Flags and fixes bugs before production

Where team AI plays

CONDITIONING
Test and QA
Writes and runs regression suites, with evidence

Where team AI plays

GROUNDS CREW
Maintain and migrate
Handles upgrades, migrations, flags and incidents

Where team AI plays

16%

of developer time is spent writing application code

IDC, 2025

42%

of the developer week lost to technical debt and bad code

Stripe, The Developer Coefficient, 2018

~83 hrs

median time for a PR to go from open to merge

LinearB, 2026

A coding assistant is a great player. An autonomous engineer is a team that covers the whole field.

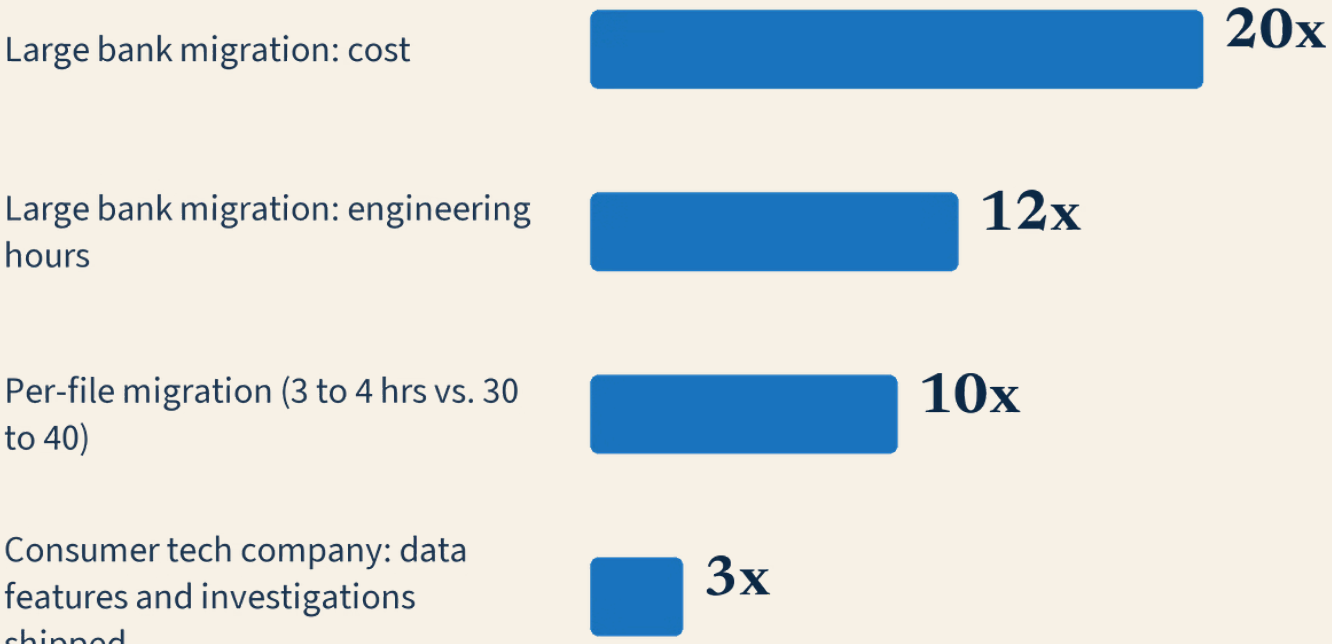
Sources: IDC, How Do Software Developers Spend Their Time? (2025); Stripe, The Developer Coefficient (2018); LinearB 2026 Benchmarks.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

AN AUTONOMOUS AI ENGINEER IN PRODUCTION AT ENTERPRISE SCALE

The scoreboard: team-level results

Improvement vs. the human-only baseline



TEST COVERAGE

50-60% → 80-90%

Typical coverage lift across customers

QUALITY ENGINEERING

93% faster

Litera regression cycles, with +40% test coverage

MIGRATIONS

1,000+ engineers' work

Major bank: ~100,000 data classes, planned as an 18-month project

DOCUMENTATION

400,000+ repos

Documented at a bank, freeing several teams for feature work

Measure AI in merged PRs, cleared backlogs and finished migrations, not suggestions accepted.

Sources: Cognition customer case studies and 2025 Performance Review. Results are customer-reported.

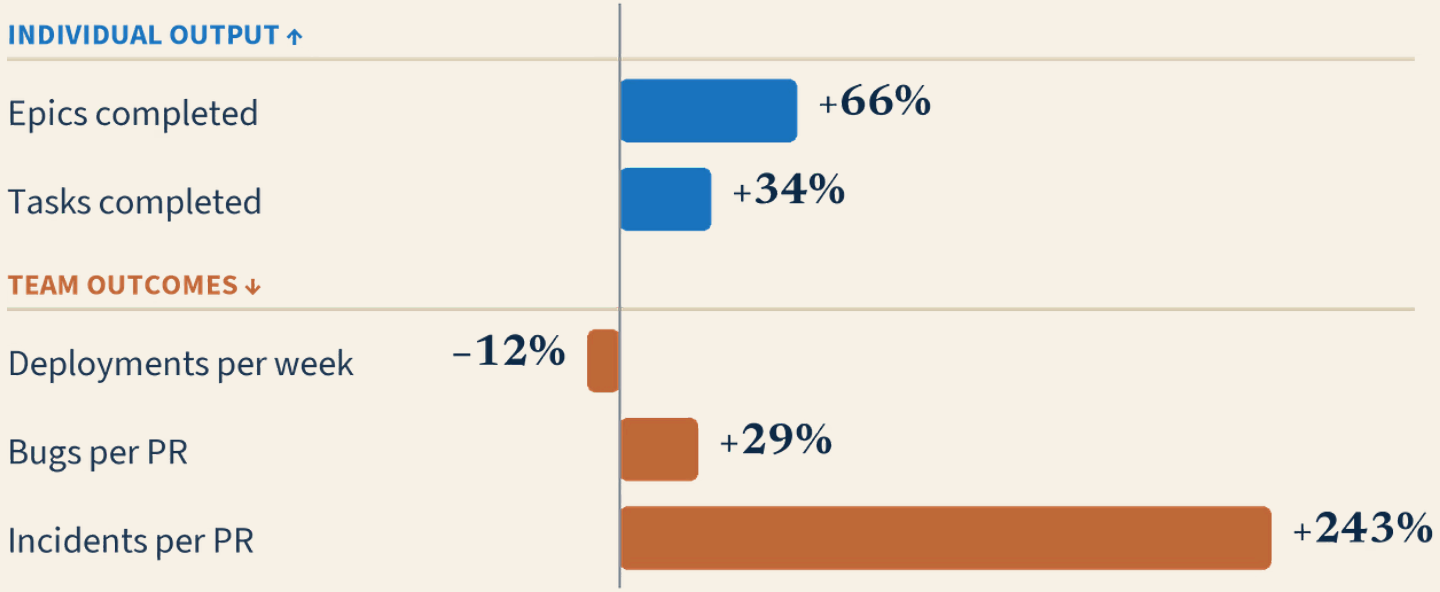
Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

THE AI PRODUCTIVITY PARADOX

Faster players don't win championships. Teams do.

What happens when only individuals get faster

High vs. low AI adoption, same organizations. Faros AI, 22K developers, 4K teams, 2026



THE BOTTLENECK MOVES DOWNSTREAM

98% more PRs merged. 91% longer review time. Company-level delivery metrics: flat.

Faros AI, 10K developers, 2025

THE QUEUE IS ALREADY LONG

Median PR takes ~83 hours from open to merge; 15 of those hours in review.

LinearB 2026 benchmark, 8.1M PRs

Coding assistants make the striker faster. The rest of the team is now the constraint.

Sources: Faros AI Engineering Report 2025 and Q2 2026; LinearB 2026 Benchmarks.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

The best teams in the world can't be better if they don't do the work to be ready.

⏪ Not ready — Ready ⏩

Test coverage and regression

Can the agent validate its own changes?

Tests should catch regressions before changes ship.

Dependency mirrors

Are OSS packages mirrored internally?

Internal mirrors keep builds working when outside sources fail.

Backend data mocking

Can it run and test without live backends?

Mocked data lets the team build and test without live systems.

Framework and language currency

How outdated is the stack?

Current frameworks and languages keep changes fast and safe.

Build and CI signals

Does CI give fast, reliable feedback?

Quick, reliable CI feedback keeps each change small and safe.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

~~Survival mode~~ → Creative mode

Software is going to flow like water · easy to move · easy to change · easy to get more of · so what's worth building changes

MAINTENANCE · KEEPS THE INSTITUTION WHERE IT IS

- [ ] Patching and upgrading what already runs
- [ ] Migrations that were due three years ago
- [ ] The vulnerability backlog
- [ ] Regulatory change, line by line
- [ ] Keeping the lights on

→

What carries the institution forward

WHAT CARRIES THE INSTITUTION FORWARD

- Digital experiences customers now expect
- Workflows rebuilt around the customer, not the system
- New products that were never 'on the roadmap'
- Resilience and controls designed in, not bolted on
- The ideas your engineers already have

Not fewer engineers. The same engineers, pointed at the things that move this institution forward.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Build software the way it should always have been built

The backlog

- Tests nobody wrote. Documentation nobody updated.
- A vulnerability open for 9 months.
- An upgrade postponed for 3 years.

Tested · Documented · Patched · Upgraded

Safe, predictable, secure.

Make AI return higher quality and lower risk, not just new features.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

OPTIMIZE FOR OUTCOMES

Optimize for business outcomes, not shiny KPIs.

Be the anti-tokenmaxxing company: focus on the metrics that move your business.

01 · USAGE

Adoption

Teams use AI across everyday engineering work.

SESSIONS · TOKENS

02 · OUTPUT

Throughput

Work you can count as it lands.

MERGED PRS · STORY POINTS

03 · TIME

Productive hours

Time saved converts into measurable engineering hours.

HOURS GAINED PER TEAM

04 · DOLLARS

P&L impact

TOP LINE

Faster shipping pulls revenue forward. Better features lift conversion.

BOTTOM LINE

More capacity without more headcount. Tools consolidated.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

CONCLUSION

How to be the AI GOAT

1. **Know your facts**
   Numbers, not narratives.
   Snarky fun facts rule at AI parties

2. **Don't follow the hype**
   Watch the slope, not the spikes.

3. **Make people justify it in business terms**
   What does it save or earn?

4. **Avoid lock-in to the frothy model game**
   Anthropic, OpenAI and SpaceX are playing it. Stay portable.

5. **Use autonomous tools, solo and in teams**
   Aim them at the whole lifecycle.

6. **Iterate constantly**
   The tools change monthly. Keep testing.

Do all six and you get a non-linear, positive, amazing impact on the businesses you own, influence, or work at.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

MONDAY MORNING

What to do next Monday

If you own an operating company

- Ask your CTO: "What have we postponed for three years?" Make one item a 90-day pilot.
- Decide the goal first: make money, save money, reduce risk, or avoid becoming obsolete.
- Measure outcomes per dollar, not tokens or seats.

If you invest

- Ask portfolio companies: How much AI spend goes to maintenance vs. new product? How is AI-written code verified?
- Code is cheap. Moats move to distribution, data, trust and workflow.
- Be skeptical of returns built on circular infrastructure financing.

If you're building

- Your edge is how fast you clear the boring 84%, not how fast you type.
- Point agents at the backlog; keep humans on taste and judgment.
- Stay portable across models.

The best AI story is the one on your P&L.

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License

Thank you

Mark Porter, Field CTO, Cognition

marklovestech.com

https://linkedin.com/in/marklovestech

Copyright (C) 2026 Mark Porter / MarkLovesTech. Apache 2.0 License