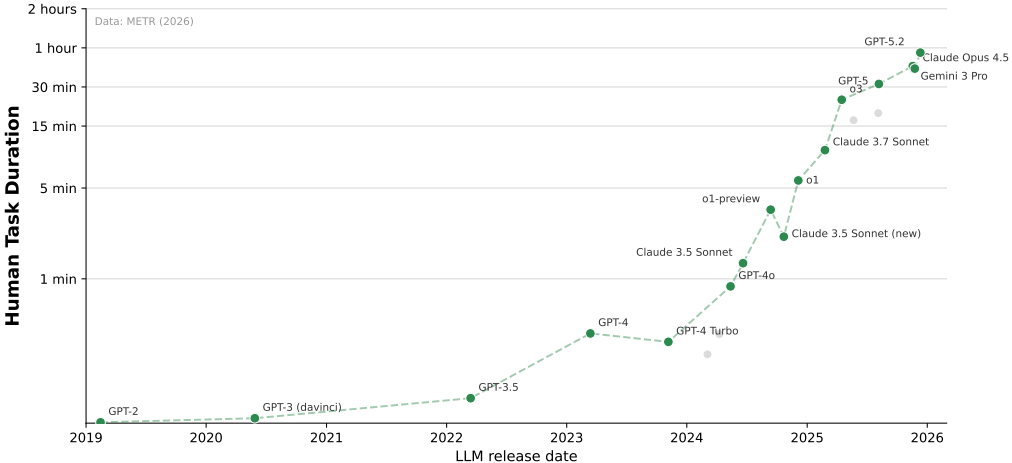
Agentic AI

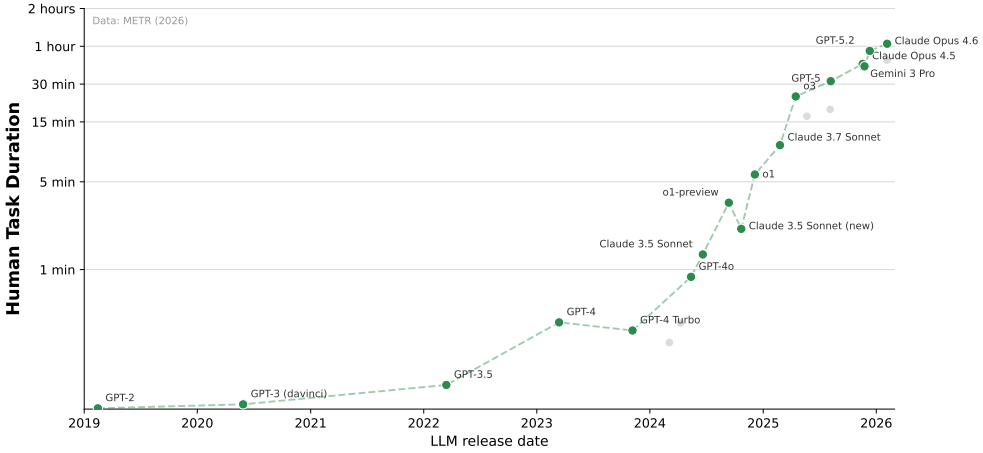
From Autocomplete to Autonomous Agents

Stefan Szeider

Faculty Retreat, February 2026


Data: METR (2026)
Human Task Duration
2 hours
1 hour
30 min
15 min
5 min
1 min
2019
2020
2021
2022
2023
2024
2025
2026
LLM release date
GPT-2
GPT-3 (davinci)
GPT-3.5
GPT-4
GPT-4 Turbo
GPT-4o
Claude 3.5 Sonnet
o1-preview
Claude 3.5 Sonnet (new)
o1
Claude 3.7 Sonnet
o3
GPT-5
Gemini 3 Pro
Claude Opus 4.5
GPT-5.2

Update: Feb 20, 2026



New: Claude Opus 4.6 and GPT-5.3 Codex — frontier crosses the 1-hour mark.

Level 1: Single-Shot LLM

Example:

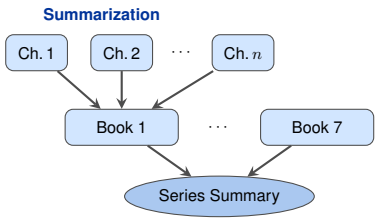
- "The cat sits on the ___" $\rightarrow$ "mat"

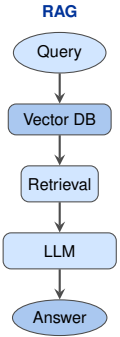
Timeline:

- 2017 — "Attention Is All You Need" (Transformer architecture)
- One input, one output — the simplest building block

arxiv.org/abs/1706.03762

Level 2: Workflows

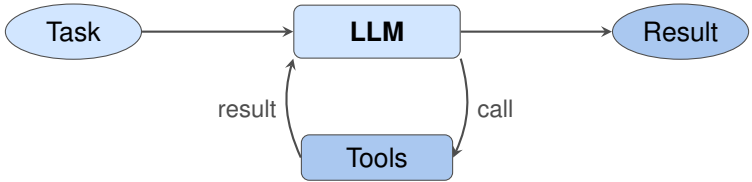




- Multiple single-shot calls stitched in a fixed sequence
- ChatGPT: chat history appended to context = workflow
- Timeline: workflows widespread from ∼2022

openai.com/index/chatgpt — arxiv.org/abs/2005.11401 (RAG)

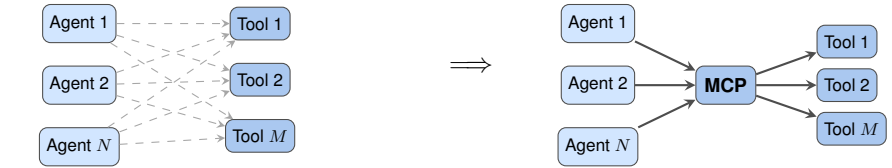
Level 3: Agentic Workflows



- The big jump: model decides *when* to call tools
- Weather lookup, arithmetic, code execution, . . .
- Opens the **neurosymbolic** world: neural LLM $\leftrightarrow$ symbolic tools
- Copilot (Level 2: completion) vs. coding agent (Level 3: write–test–fix loop)
- Tool/function calling introduced $\sim$2023 (GPT-4 era)

openai.com/index/function-calling (2023) — github.com/features/copilot

Model Context Protocol (MCP)



- “The USB port for tool calling” — introduced late 2024
- Thousands of MCP servers: GitHub, AWS, Google Workspace, Perplexity, . . .

modelcontextprotocol.io — anthropic.com/news/model-context-protocol

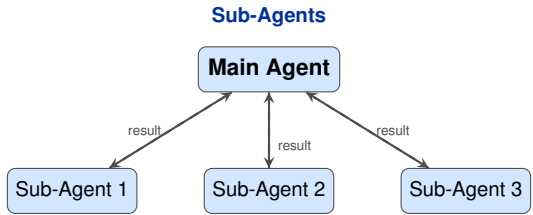
Our MCP Servers

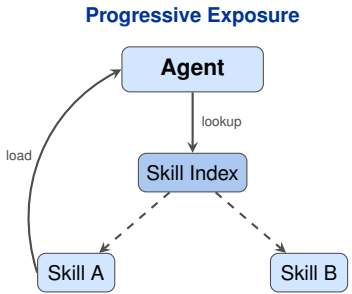
Server	Function	Publication
`MCP-Solver`	SAT / CP / SMT solving	SAT'25
`DBLP-MCP`	unmediated literature research	AI4SC@AAAI'26
`Consult-7`	synthesize feedback from frontier models	PyPI
`iPython-MCP`	virtual Python coding environment	AGENT@ICSE'26
`Sage-MCP`	computer algebra system	internal
`Lean-MCP`	proofs in Lean 4 system	internal

- Each server exposes symbolic tools to the LLM via MCP
- Agent autonomously decides when and how to call them

github.com/szeider/mcp-solver — github.com/szeider/mcp-dblp — github.com/szeider/consult7 — github.com/szeider/agentic-python-coder

Level 4: Context Engineering

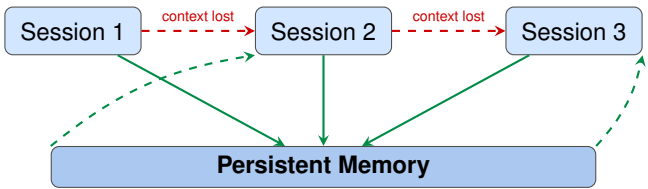




- Context window gets exhausted during long agent runs
- **Sub-agents**: encapsulated context, only results bubble up
- **Skills**: short index loaded upfront, full instructions on demand

anthropic.com/research/building-effective-agents

Level 5: Persistent Memory / Self-Improvement



- Session context is ephemeral — gone when session ends
- Persistent memory bridges sessions $\rightarrow$ agents improve over time
- Agents can generate their own skills $\rightarrow$ self-improvement loop
- Claude Code: 100% of its own code is written by itself
- 100K-line C compiler: 16 Claude agents, 2 weeks

fortune.com/2026/01/29/100-percent-of-code-at-anthropic-and-openai-is-now-ai-written — anthropic.com/engineering/building-c-compiler

OpenClaw

Created by: Peter Steinberger (TU Wien alumnus)

Key facts:

- Released Nov 2025 — 196k GitHub stars, 34k forks
- 900k weekly npm downloads, 2M visitors in first week
- Assembles existing agentic techniques into one package
- Persistent memory, browser automation, heartbeat triggers

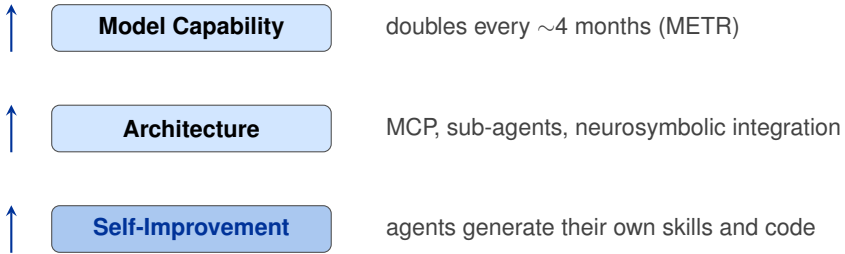
Caveat:

- Very insecure — user gives full control to the agent

Not fundamentally new — but the right combination at the right time.

github.com/openclaw/openclaw

Three Scaling Dimensions

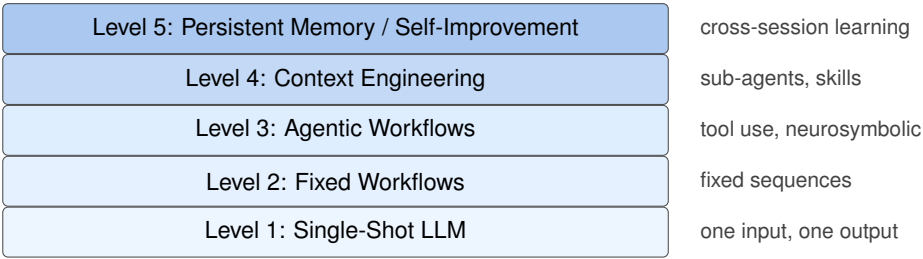


Implication: software will become cheap or free.

When people discuss the singularity — this is where it starts.

metr.org/time-horizons

Summary



Agentic AI

From Autocomplete to Autonomous Agents