

Intelligence Everywhere: 12 Months That Redefined AI and Internet Design

Tony Kay, AI Lead, Red Hat
tok@redhat.com



- AI Lead, Red Hat Portfolio Technology
- RITE 2024 Speaker
 - Thanks for having me back!
- Passionate about
 - Spec Coding
 - AI Engineering
 - Agentic AI
 - Inference and LLMs



Large Language Models (LLMs)



AI

Gemini



ERICSSON



- Bigger models, but also, smaller models
 - High Quality Training
 - Better Quantization
- Better Reasoning
- Better Tool Calling
- More and better Open Source Models
- More Speciality Models including:
 - Anomaly Detection, Predictive Maintenance
 - Churn Prediction, Network Optimization
 - Capacity Planning



China Rising



- DeepSeek-R1 and V3 Series:
 - V3 and V3.1 hybrid “Thinking/Non-Thinking”
 - Caused the “Deep Seek Moment”
- Alibaba Qwen3 family (Apache 2 License):
 - Up to 235B and 480B Qwen3-Coder
- Moonshot Kimi-K2, Baidu ERNIE, and more
- Many Open Source contributions
- Huawei and ZTE
 - Telecom Foundation Model (2024)
 - Various AI Carrier products shipping



Shift: Prompt Engineering -> Context Engineering

Prompt Engineering vs. Context Engineering

Understanding the evolution from single-turn instruction optimization to multi-turn agent state management.



Prompt Engineering

Definition: Methods for writing and organizing initial LLM instructions (prompts) for optimal outcomes.

↳ Primary Focus (Single Turn)

- How to write effective System Prompts.
- Optimization for One-shot Classification or static generation tasks.
- The initial instruction sent to the model.

→
PROGRESSION



Context Engineering

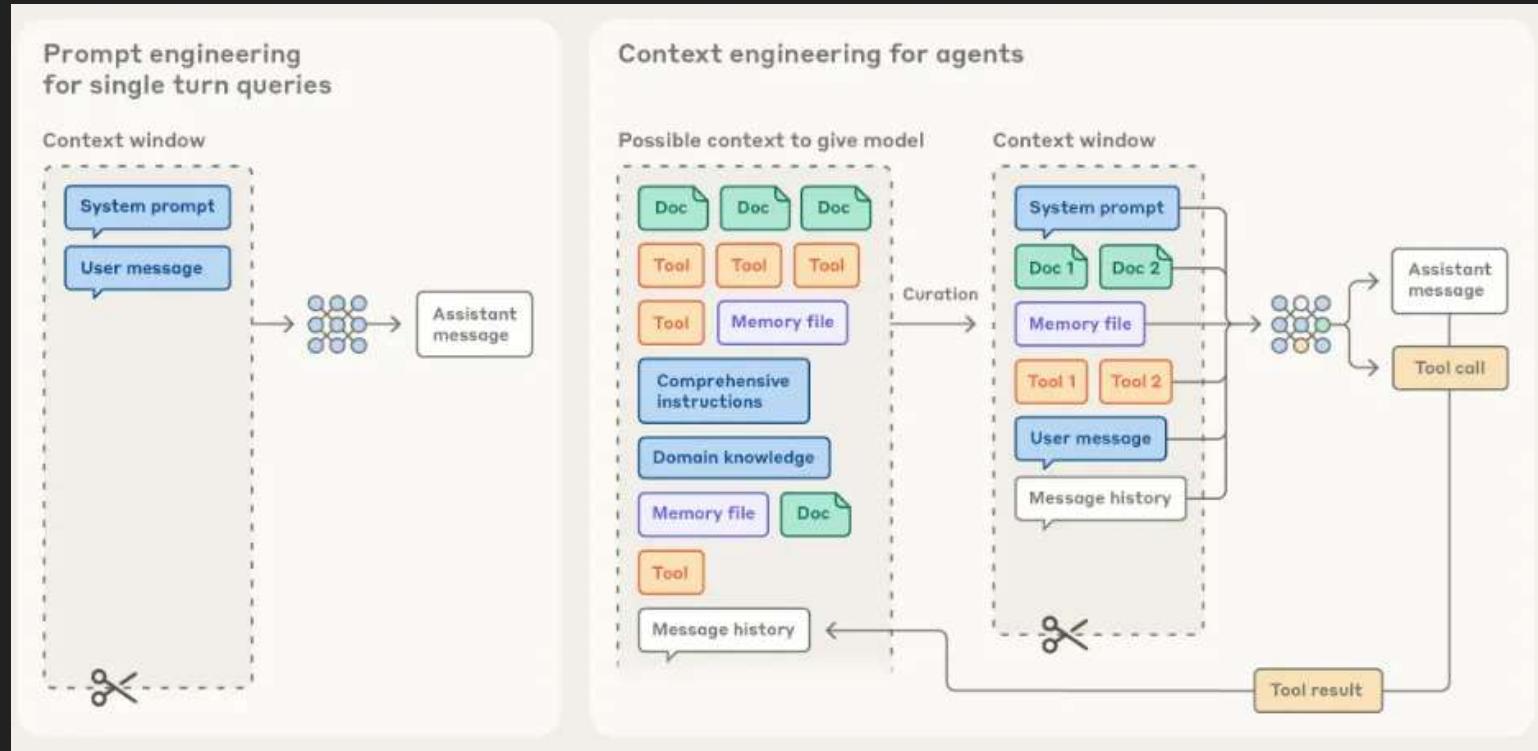
Definition: Strategies for curating and maintaining the optimal set of tokens (information) across multi-turn inference.

↳ Primary Focus (Multi-Turn / Long Horizon)

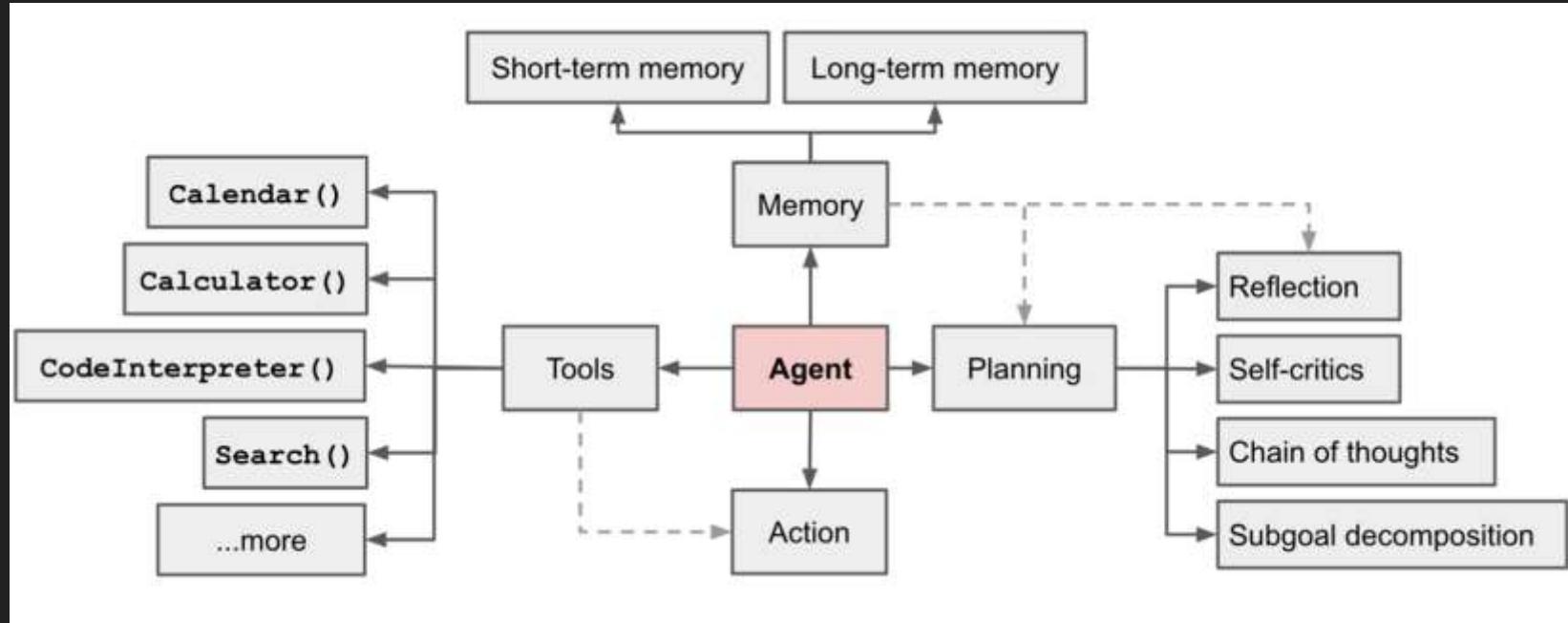
- Managing the entire Context State (history, tools, external data, etc.).
- Cyclically refining information for agents running in a Continuous Loop.
- Curating data to fit the Limited Context Window.



Shift: Prompt Engineering -> Context Engineering



Agents - “2025 - the year of Agents”



Agents

- Models globally became stronger at
 - Reasoning
 - Tool Calling
- Anthropic and OpenAI follow Perplexity's Agentic Lead
- Chinese Agentic Assistant Manus 1 and Manus 1.5 Released
- Frameworks and Frameworkless evolved
 - LangGraph and LangChain went 1.0
 - Pydantic AI, Smolagents, OpenAI Agent SDK released
- Agents became “deeper”
 - LangChain releases Deep Agents



DeepAgents Core Capabilities

Building LLM Agents that can plan, remember, and manage complexity.



1. Planning & Task Breakdown

DeepAgents utilize the built-in `write.todos` tool to break large, ambiguous objectives into sequential, manageable steps. This allows them to track progress and dynamically adjust the plan as new information is gathered, moving beyond shallow, immediate action.



2. Context Management

By leveraging file tools (`ls`, `read_file`, etc.), agents can offload large amounts of data, code, or research notes from the LLM's short-term memory. This prevents context window overflow, enabling the smooth handling of highly detailed and lengthy tasks.



3. Sub-Agent Delegation

The powerful `task` tool enables the main agent to delegate specific, complex parts of the problem to specialized sub-agents. This modular approach minimizes context clutter for the primary agent and ensures focused, efficient execution of sub-tasks.



4. Long-Term Memory

Integrated with LangGraph's `Store`, these agents can remember context, decisions, and outcomes across multiple sessions. This crucial capability allows them to build on past work, resume interrupted conversations, and develop a persistent, growing knowledge base.



Model Context Protocol (MCP)

- Announced by Anthropic in November 2024, Open Standard
 - Achieved widespread adoption including by rivals
- Quickly over 1,000 MCP Servers by February 2025
- MCP Servers:
 - Tools - executable server-side functions
 - Resources are data objects or endpoints
 - Structured data, or SQL query etc
 - Prompts - reusable templated messages or instructions
- Security remains a **huge concern**

Vibe Coding, Spec Coding



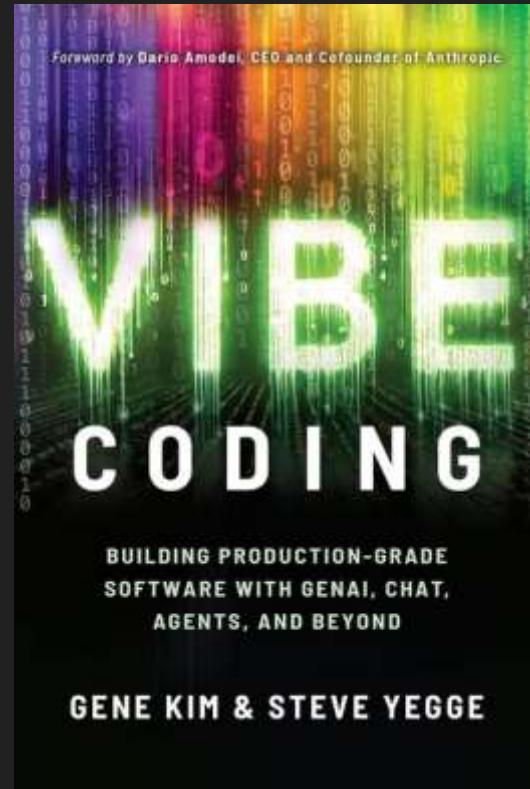
Andrej Karpathy

@karpathy

0 ...

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. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

4:17 PM · Feb 2, 2025 · 5M Views



- Agentic Coding Tools driven by formal Specifications
- Frameworks and Methodologies evolving
 - **spec-kit** GitHub/Microsoft
 - Released early September, 79 releases, 41K stars
 - AWS Kiro - Codeserver based IDE, “Spec First”
 - Multiple new “Spec Frameworks”
 - Also many evolving Spec practices



So what's coming next for 2026?

- Autonomous AI and Agentic Systems
 - Agentic Networking and Network Engineers?
- Open Source Frontier Models
 - Even higher quality “Small” < 100B parameter models
- Wide scale Spec Coding?
 - What will this look like for Networking?
- Will we get serious about AI Security?

Invitation: I'll be outside the session. Come out and we can make some Agentic Networking and other Engineers together



Thank you!