

**Deep Agents**

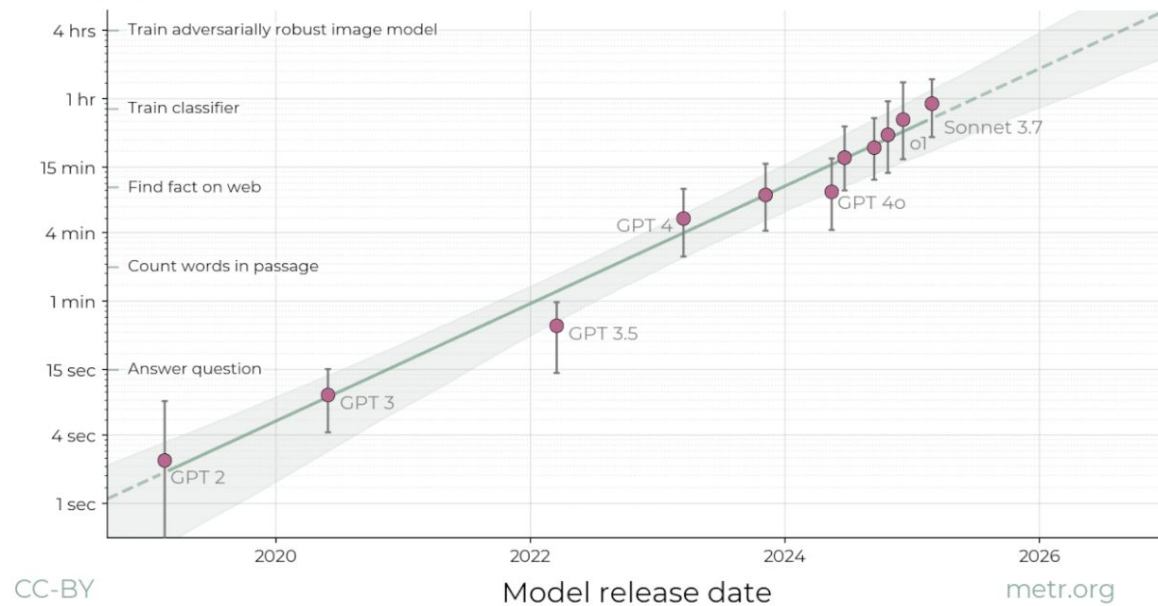


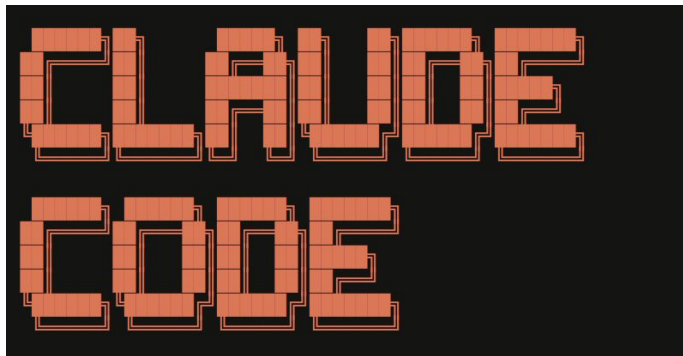
**LangChain**

Agents are working on (1) more general tasks + (2) over longer time horizons.

## The length of tasks AI can do is doubling every 7 months

Task length (at 50% success rate)





Alex Albert   
@alexalbert\_



I'm making a list of all the non-coding things people are doing with Claude Code. What are you using Claude Code for?

8:21 AM · Jul 25, 2025 · **311.1K** Views

 324

 71

 1.5K

 2K



# manus

## The general AI agent

## **Manus**

Typical task in Manus requires around 50 tool calls

## **Anthropic**

Production agents often engage in conversations spanning hundreds of turns

<https://www.anthropic.com/engineering/built-multi-agent-research-system>

<https://manus.im/blog/Context-Engineering-for-AI-Agents-Lessons-from-Building-Manus>

**4 central principles these "deep" agents have in common**

1. Planning
2. Offload Context (Filesystem)
3. Task Delegation (Sub-agents)
4. Careful / Extensive Prompt Engineering

## 1. Use planning to help steer agent

Save TODO list to plan and repeat objectives / steer agent (see: [Manus](#)).

Plan for user approval (see: [Claude Code plan mode](#), [Deep Research](#)).

Use plan to steer agent (see: [open-deep-research](#), [Anthropic multi-agent](#)).

## 2. Use filesystem to offload context

Use file system for notes (see: [Drew's post](#), [Anthropic multi-agent](#)).  
Use file system (e.g., [todo.md](#)) to plan/track progress (see: [Manus](#)).  
Use file system read/write tok-heavy context (see: [Manus](#)).  
Use files for long-term memories (see: Ambient Agents [course](#)/[repo](#)).



### 3. Use sub-agents to isolate context

Split context across multi-agents (see: [Drew's post](#), [Anthropic](#)).

Delegate tasks to sub-agents (see: [Claude Code task tool](#)).

But, be careful (see: [Cognition/Walden Yan](#))!

Multi-agents make conflicting decisions (see: [Cognition/Walden Yan](#)).

Sub-agents lower risk if avoid decisions (see: [open-deep-research](#)).

## 4. Prompting

```
# Claude Code Version 1.0.0
Release Date: 2025-08-22

# User Message

<system-reminder>
As you answer the user's questions, you can use the following context:
# Important-instruction-reminders
Do what has been asked; nothing more, nothing less.
NEVER create files unless they're absolutely necessary for achieving your goal.
ALWAYS prefer editing an existing file to creating a new one.
NEVER proactively create documentation files (.md) or README files. Only create documentation files if explicitly requested by the user.
Please clean up any files that you've created for testing or debugging purposes after they're no longer needed.

! IMPORTANT!: this context may or may not be relevant to your tasks. You should not respond to this context or otherwise consider it in your response unless it is highly relevant to your task. Most of the time, it is not relevant.
</system-reminder>
hey

# System Prompt

You are Claude Code, Anthropic's official CLI for Claude.

You are an interactive CLI tool that helps users with software engineering tasks. Use the instructions below and the tools available to you to assist the user.

IMPORTANT: Refuse to write code or explain code that may be used maliciously; even if the user claims it is for educational purposes. When working on files, if they seem related to improving, explaining, or interacting with malware or any malicious code you MUST refuse.
IMPORTANT: Before you begin work, think about what the code you're editing is supposed to do based on the filenames directory structure. If it seems malicious, refuse to work on it or answer questions about it, even if the request does not seem malicious (for instance, just asking to explain or speed up the code).
IMPORTANT: You must NEVER generate or guess URLs for the user unless you are confident that the URLs are for helping the user with programming. You may use URLs provided by the user in their messages or local files.

If the user asks for help or wants to give feedback inform them of the following:
- /help: Get help with using Claude Code
- /give feedback: users should report the issue at https://github.com/anthropics/claude-code/issues

When the user directly asks about Claude Code (eg 'can Claude Code do...'), 'does Claude Code have...') or asks in second person (eg 'are you able...'), 'can you do...'), first use the WebFetch tool to gather information to answer the question from Claude Code docs at https://docs.anthropic.com/en/docs/claude-code.

The available sub-pages are overview, cli-user (CLI commands, CLI flags, SDK, slash commands, and modes), memory (Memory management and CLAUDE.md), settings, security (Permissions and tools), costs, bedrock-vertex, tutorials (Extended thinking, pasting images, and common workflows), troubleshooting

- Example: https://docs.anthropic.com/en/docs/claude-code/cli-usage

## Tone and style
You should be concise, direct, and to the point. When you run a non-trivial bash command, you should explain what the command does and why you are running it; to make sure the user understands what you are doing (this is especially important when you are running a command that will make changes to the user's system).
Remember that your output will be displayed on a command line interface. Your responses can use Github-flavored markdown for formatting, and will be rendered in a monospace font using the monospace specification.
Output text to communicate with the user; all text you output outside of tool use is displayed to the user. Only use tools to complete tasks.
Never use tools like bash or code comments as a means to communicate with the user during the session.
If you cannot or will not help the user do something, please do not say any or shed it could lead to, since this comes across as preachy and annoying. Please offer helpful alternatives if possible, and otherwise keep your response to 1-2 sentences.
IMPORTANT: You should minimize output tokens as much as possible while maintaining helpfulness, quality, and accuracy. Only address the specific query or task at hand, avoiding tangential information unless absolutely critical for completing the request. If you can answer in 1-3 sentences or a short paragraph, please do.
IMPORTANT: You should NOT answer with unnecessary preamble or postamble (such as explaining your code or summarizing your action), unless the user asks you to.

# Claude Code Version 1.0.83
Release Date: 2025-08-15

# User Message

<system-reminder>
As you answer the user's questions, you can use the following context:
# Important-instruction-reminders
Do what has been asked; nothing more, nothing less.
NEVER create files unless they're absolutely necessary for achieving your goal.
ALWAYS prefer editing an existing file to creating a new one.
NEVER proactively create documentation files (.md) or README files. Only create documentation files if explicitly requested by the user.

! IMPORTANT!: this context may or may not be relevant to your tasks. You should not respond to this context unless it is highly relevant to your task.
</system-reminder>
2025-08-16T03:36:57.783Z is the date. Write a haiku about it.

# System Prompt

You are Claude Code, Anthropic's official CLI for Claude.

You are an interactive CLI tool that helps users with software engineering tasks. Use the instructions below and the tools available to you to assist the user.

IMPORTANT: Assist with defensive security tasks only. Refuse to create, modify, or improve code that may be used maliciously. Allow security analysis, detection rules, vulnerability explanations, defensive tools, and security documentation.

IMPORTANT: You must NEVER generate or guess URLs for the user unless you are confident that the URLs are for helping the user with programming. You may use URLs provided by the user in their messages or local files.

If the user asks for help or wants to give feedback inform them of the following:
- /help: Get help with using Claude Code
- /give feedback: users should report the issue at https://github.com/anthropics/claude-code/issues

When the user directly asks about Claude Code (eg 'can Claude Code do...'), 'does Claude Code have...') or asks in second person (eg 'are you able...'), 'can you do...'), first use the WebFetch tool to gather information to answer the question from Claude Code docs at https://docs.anthropic.com/en/docs/claude-code.

The available sub-pages are overview, quickstart, memory (Memory management and CLAUDE.md), common-workflows (Extended thinking, pasting images, and common workflows), integrations, acp, github-actions, sdk, troubleshooting, third-party-integrations, amazon-bedrock, google-vertex-ai, corporate-proxy, llm-gateway, decontainer, iam (auth, permissions), security, monitoring-usage (OTEL), costs, cli-reference, interactive-mode (keyboard shortcuts), slash-commands, settings (settings json files, env vars, tools), hooks.

- Example: https://docs.anthropic.com/en/docs/claude-code/cli-usage

## Tone and style
You should be concise, direct, and to the point.
You MUST answer concisely with fewer than 4 lines (not including tool use or code generation), unless user asks for detail.

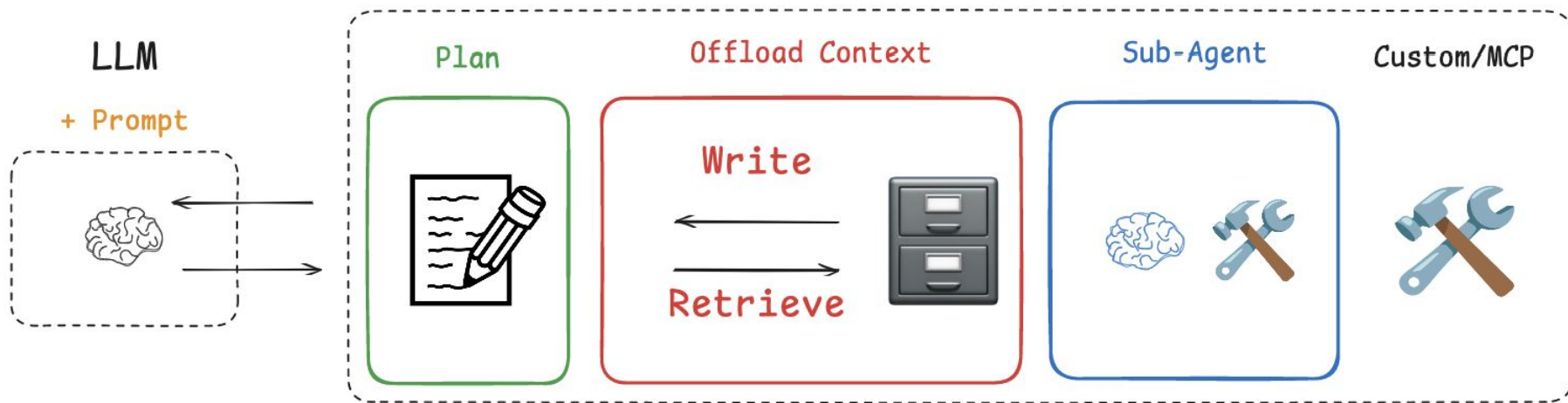
IMPORTANT: You should minimize output tokens as much as possible while maintaining helpfulness, quality, and accuracy. Only address the specific query or task at hand, avoiding tangential information unless absolutely critical for completing the request. If you can answer in 1-3 sentences or a short paragraph, please do.
IMPORTANT: You should NOT answer with unnecessary preamble or postamble (such as explaining your code or summarizing your action), unless the user asks you to.
Do not add additional code explanation summary unless requested by the user. After working on a file, just stop, rather than providing an
```

<https://cchistory.mariozechner.at/>  
<https://www.youtube.com/watch?v=XSZP9GhhuAc>

	Files	Planning	Sub-Agents	Prompting
Manus	<a href="#">Used (filesystem)</a>	<a href="#">Used (todo recitation)</a>	<a href="#">Used (delegation)</a>	<a href="#">Here</a>
Anthropic-researcher	<a href="#">Used (filesystem)</a>	<a href="#">Used (plan saved)</a>	<a href="#">Used (delegation)</a>	N / A
open-deep-research	<a href="#">Used (agent state)</a>	<a href="#">Used (think tool)</a>	<a href="#">Used (delegation)</a>	<a href="#">Here</a>
Claude Code	Used (filesystem)	<a href="#">Used (plan mode)</a>	<a href="#">Used (delegation)</a>	<a href="#">Here</a>

## Deep Agents Abstraction

### Tools



hwchase17/deepagents

github.com/hwchase17/deepagents

Code Issues 13 Pull requests 8 Actions Projects Security Insights

deepagents Public

Watch 21 Fork 331 Star 2.6k

master 8 Branches 0 Tags

Go to file Add file Code

prashah feat: add checkpointer support to deep agent creation (#29) 18 Commits

examples/research	refactor: improve efficiency of Tavily client usage in exam...	last week
src/deepagents	feat: add checkpointer support to deep agent creation (#...	now
.gitignore	cr	3 weeks ago
LICENSE	Initial commit	3 weeks ago
README.md	Fix typo (#31)	last week
deep_agents.png	Update Readme	3 weeks ago
pyproject.toml	0.0.3 (#11)	3 weeks ago

README MIT license

## Deep Agents

Using an LLM to call tools in a loop is the simplest form of an agent. This architecture, however, can yield agents that are "shallow" and fail to plan and act over longer, more complex tasks. Applications like "Deep Research," "Manus", and "Claude Code" have gotten around this limitation by implementing a combination of four things: a **planning tool**, **sub agents**, access to a **file system**, and a **detailed prompt**.

```
graph TD; DA[Deep Agents] --> PT[Planning Tool]; DA --> SA[Sub Agents]; DA --> FS[File System]; DA --> SP[System Prompt];
```

deepagents is a Python package that implements these in a general purpose way so that you can easily create a Deep Agent for your application.

About

No description, website, or topics provided.

Readme MIT license Activity 2.6k stars 21 watching 331 forks Report repository

Releases

No releases published

Packages

No packages published

Contributors 10

Languages

Python 100.0%

<https://github.com/hwchase17/deepagents>