DecompileD Masterclass 2026

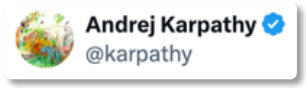
Spec-Driven Development

Markus Höfling

“**Vibes** ship demos.

Specs ship products.”

Vibe Coding



- **“forget that the code even exists”**
- “accept all always”
- “don’t read the diffs”
- “copy & paste error messages”
- often associated with non-programmers now starting to code

Agentic Coding

- building software using **coding agents**
- can generate & execute code
- iteratively test and fix code
- turn-by-turn guidance by user
- done by actual software devs trying to accelerate

Application of Coding Agents

Scope of this Masterclass

devboost.com/blog

devboost.com/ai-report

- "**loop with tools**" orchestration pattern
- relies on its **own judgement**
- determined by the system prompt, system tools, ..

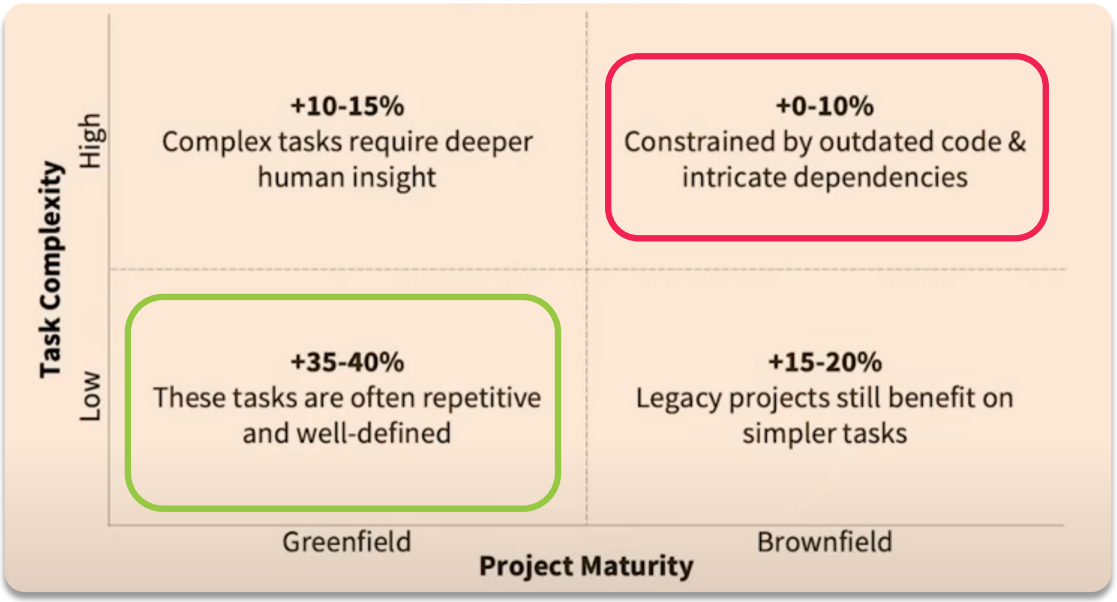
background-agents.com/

Software Development Lifecycle

Limitations of Coding Agents

Greenfield vs. Brownfield Success

Dev's productivity gains from the use of AI



Yegor Denisov-Blanch on Developer Productivty | Stanford

How is this addressed?

Context Engineering

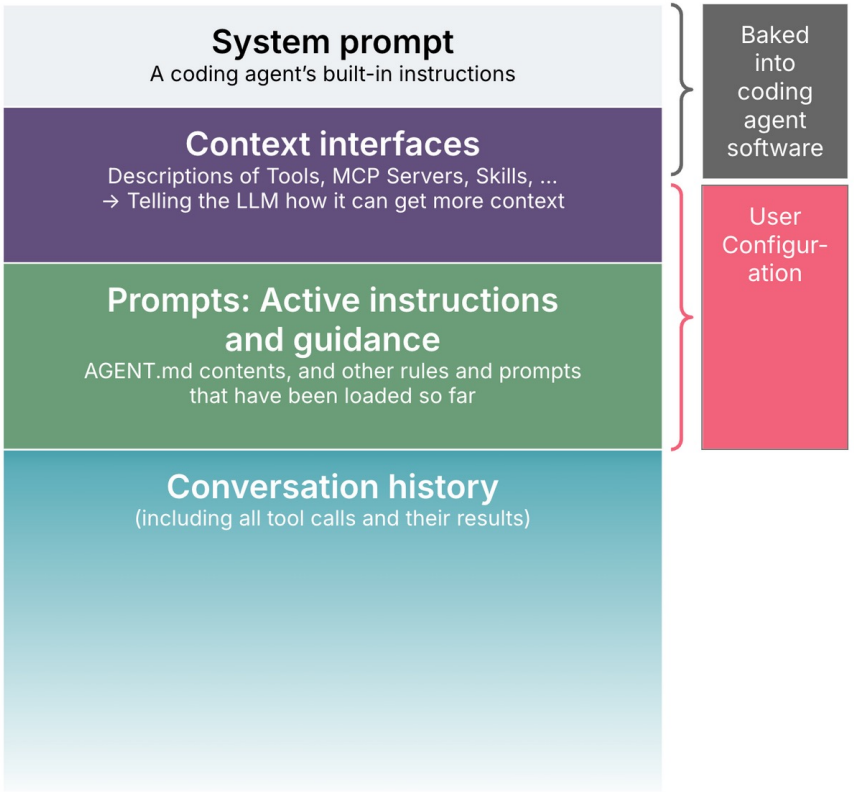
The Coding Agent's Core Value

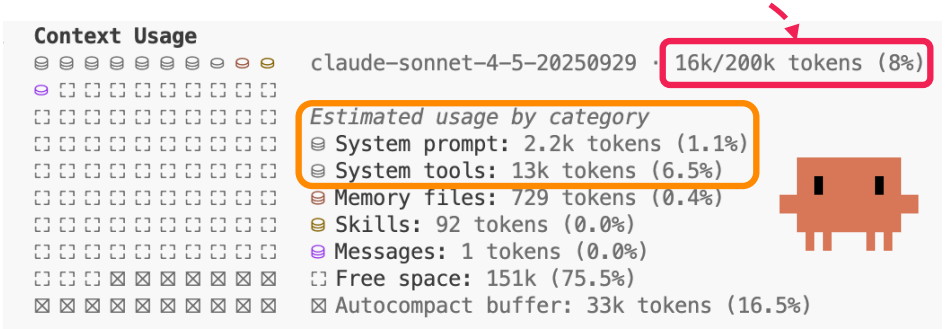
Context Engineering

Context Engineering

"Context engineering is the *systematic design* and **optimization of the information provided to a LLM** during inference to **reliably** produce the desired **output**."

https://www.thoughtworks.com/radar/techniques/context-engineering



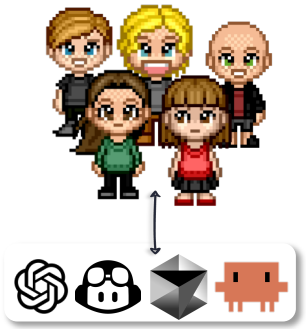


"Read, Edit, Write, Bash, Glob, LS, ..."

martinfowler.com/articles/exploring-gen-ai/context-engineering-coding-agents.html

Limitations of Coding Agents

Human Intent and Judgment



"AI is good at searching solution spaces. It is *not good* at **deciding which problems matter**, what **trade-offs are acceptable**, or when something is **correct enough to ship**. Those **decisions remain human**, […] ."

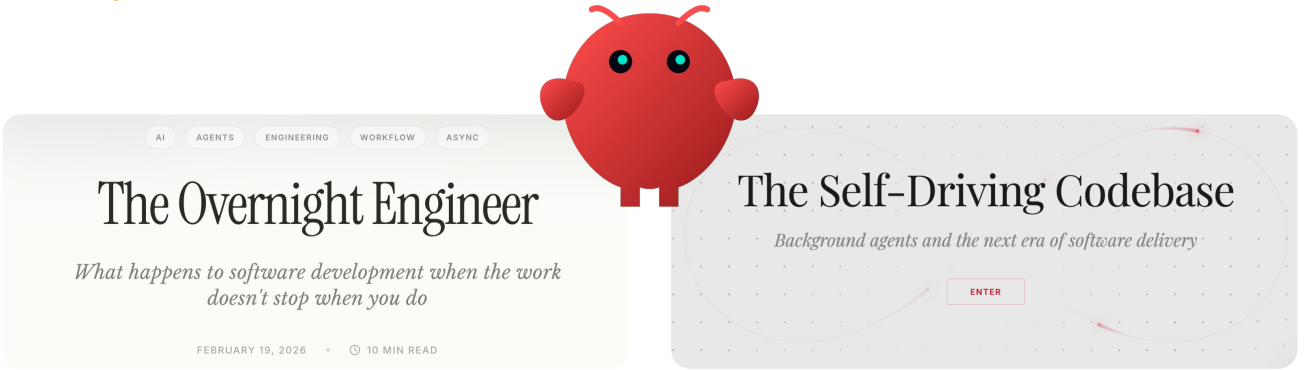
ona.com/stories/industrializing-software-development

Software Development Lifecycle

background-agents.com/

Outlook: Background Agents

Highest Level of Autonomy



What this requires:

- **Context curation** … Agents work best with **clean**, **focused** and **explicit** context
- **Decomposition** … **break work into chunks** that can be completed **independently**
- **Specification** … the **WHY over the WHAT** in precise, testable acceptance criteria

Roast
STRUCTURED AI WORKFLOWS

Honk

Minions

Inspect

Context Engineering

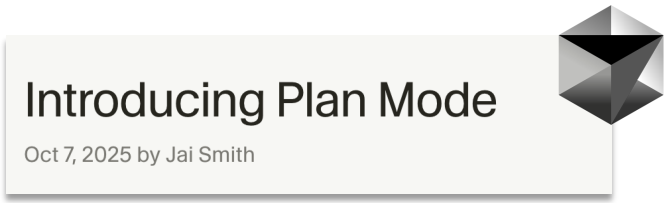
..with Spec-Driven Development

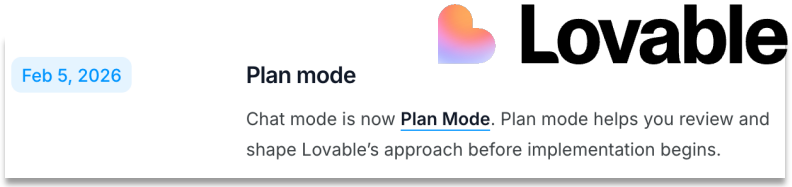
Coding Agents

Spec-Driven Development

- **writing a "spec" before writing code** which makes **context explicit**
- spec becomes **source of alignment** (**contract**) between human & agent
- helps to achieve **clarity** and building **mental model**
- **enforces context engineering** under the hood
- makes iteration and collaboration easier
- helps to become more model & tool agnostic
  (*you* define the workflow, not vice versa!)

… vs. plan / act:





Spec-Driven Development

Tooling

OPENSPEC
(by Fission AI)

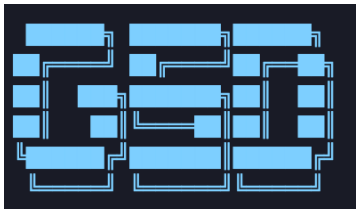
Conductor

Spec Kit
(by Github)

BMAD METHOD

odeplain

Tessl



Kiro.dev
(by Amazon)

Spec-Driven Development

Conductor

Conductor

- **Conductor** has been originally populated as plugin for the Gemini CLI
- The instructions & templates are open source -> now available **for Claude Code as plugins** too! 🎉
- The plugin consists out of **Skills & Templates** for Claude Code and offers a predefined workflow:

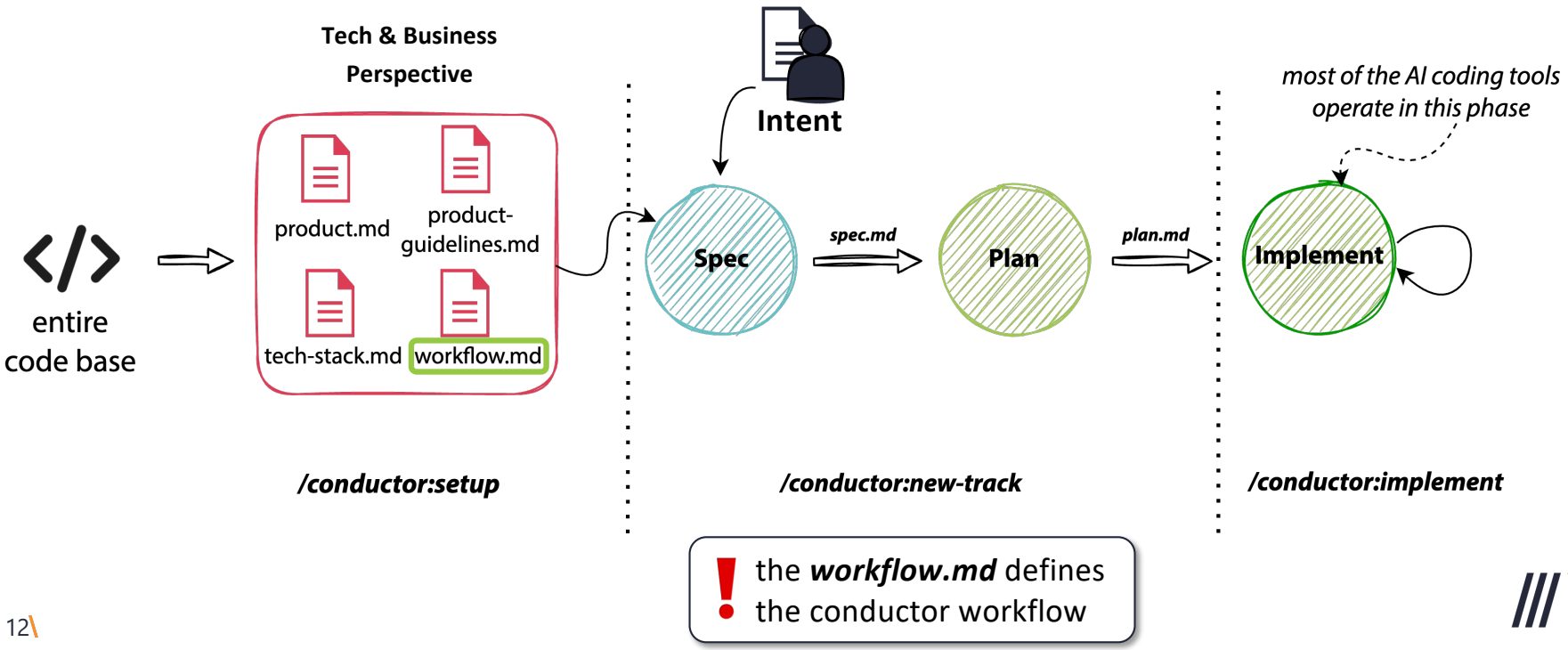
Context -> Spec & Plan -> Implement

```
# in Claude Code
/conductor:setup          - Initialize project context and scaffolding
/conductor:new-track      - Create a new track with spec and plan
/conductor:implement      - Execute tasks from the current track
/conductor:status         - Summarize current progress
/conductor:revert         - Git-aware revert of tracks, phases, or tasks
```

Spec-Driven Development

Conductor | Workflow

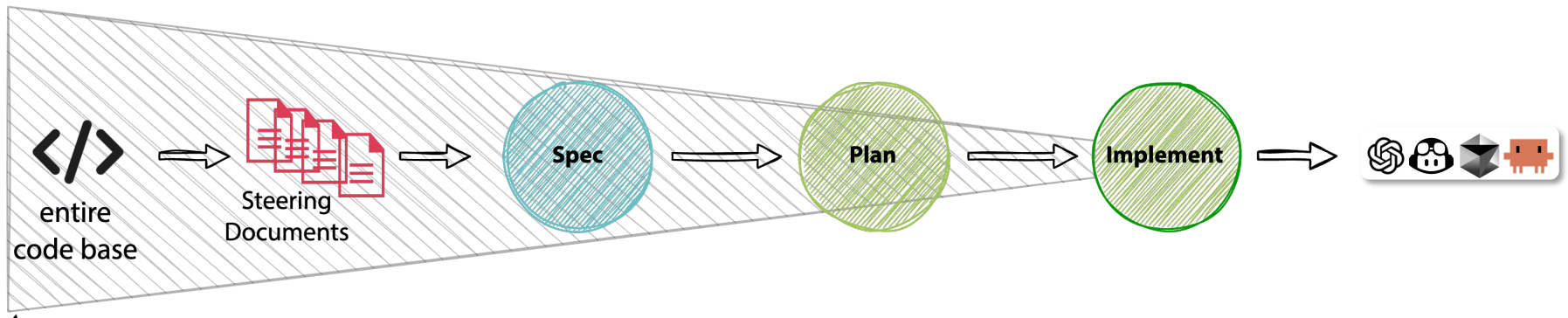
Conductor



Spec-Driven Development

Conductor | Context Curation

Conductor



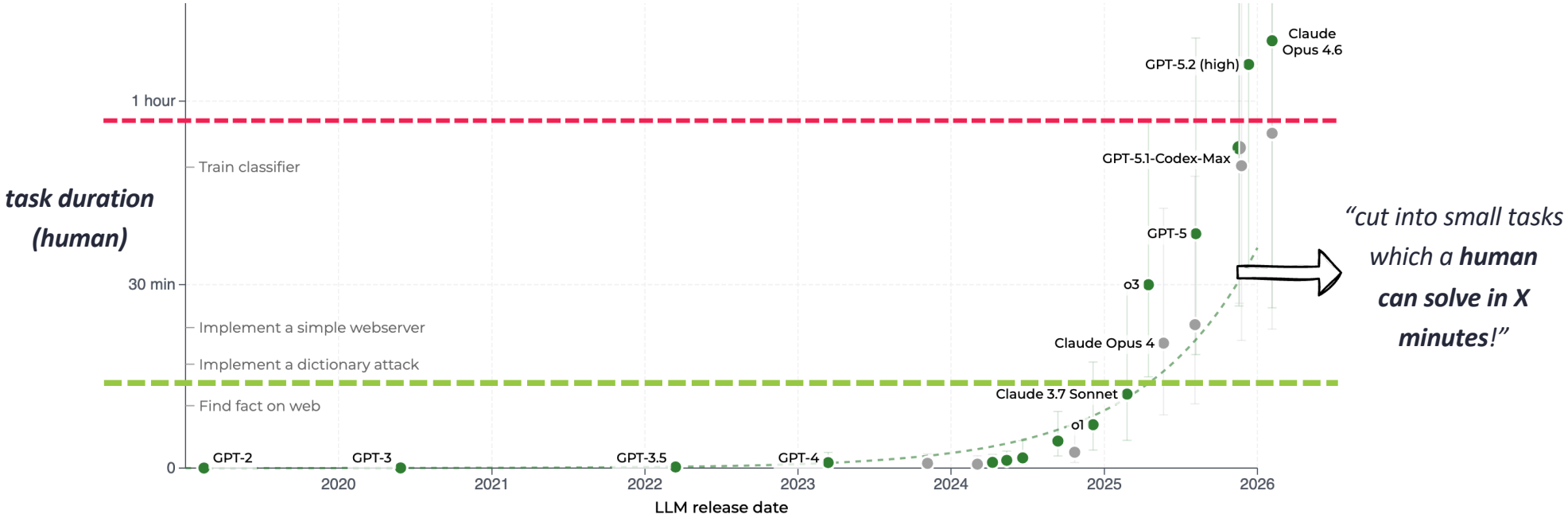
- **Context size**
- **Solution space**
- **Task Duration**

Context Engineering

Subtask Duration

METR

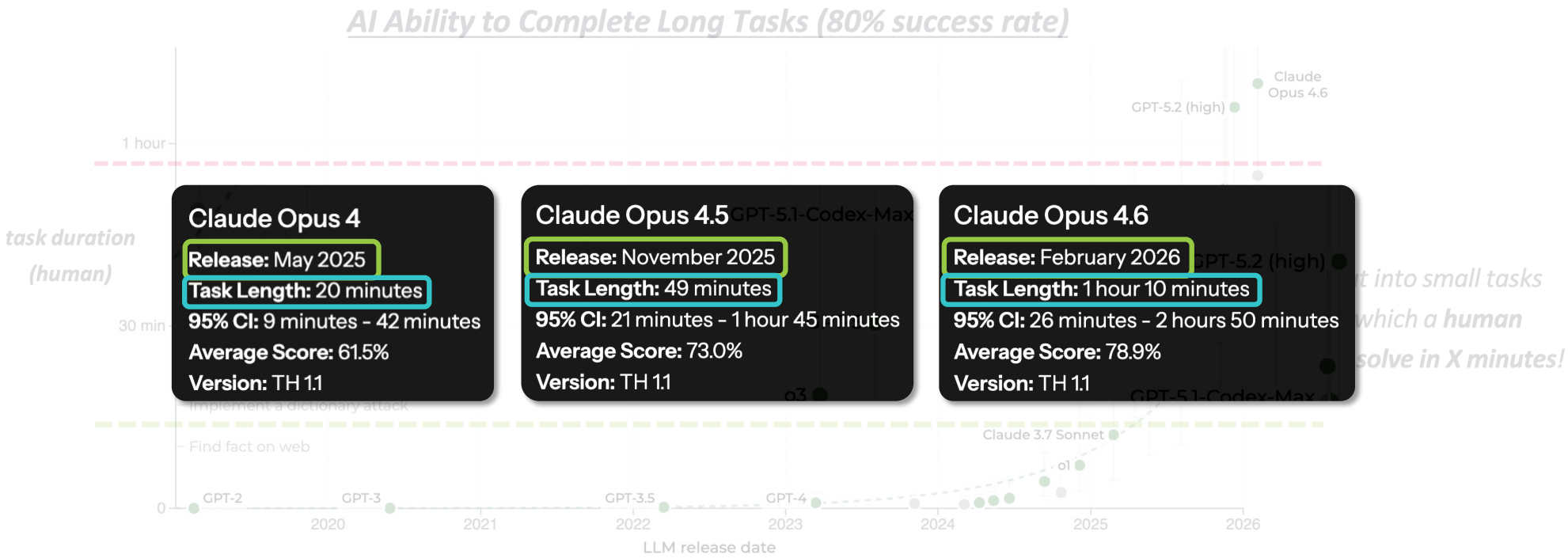
AI Ability to Complete Long Tasks (80% success rate)



https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/

Context Engineering

Subtask Duration



https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/

Hands-On

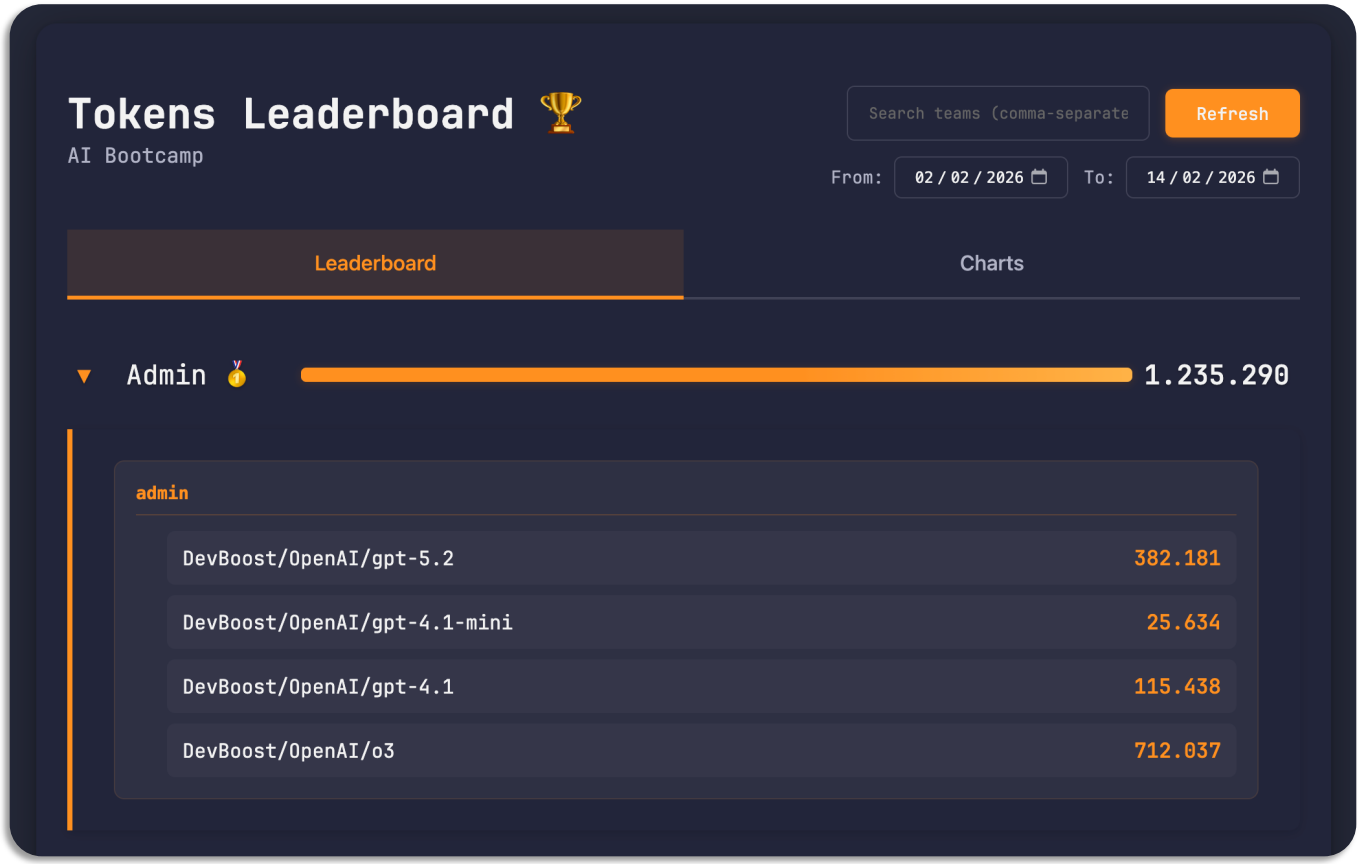
Spec-Driven Development | Leaderboard Demo

Spec-Driven Development

Leaderboard | Behind the Scenes

Spec-Driven Development

Leaderboard | Demo Project



Spec-Driven Development

Leaderboard | Tooling

Task:
Implement backend route

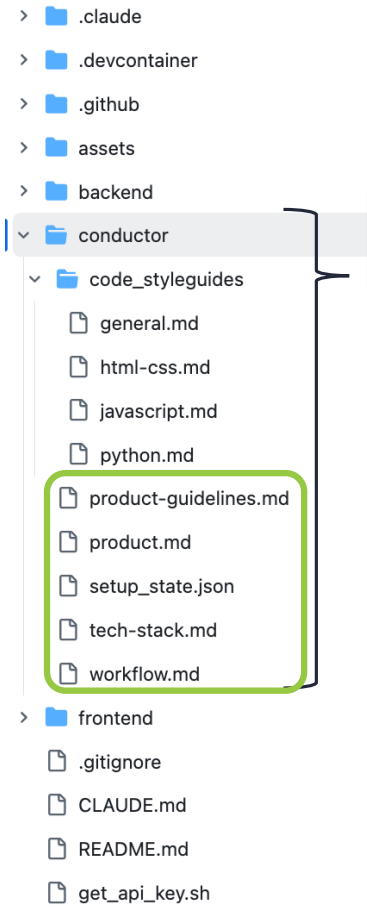
Conductor

- Claude Code is one of the **leading AI Coding Tools**
- **Claude Code Router (CCR)** acts as a proxy and allows further endpoints

- **Conductor** has been originally populated as plugin for the Gemini CLI
- **Approach: Context -> Spec & Plan -> Implement**

Spec-Driven Development

Leaderboard



The project is already "setup"!

```
/conductor:setup        - Initialize project context and scaffolding
```

```
2.1 Product Guide — Question 1

Question type: Additive

Who are the primary users of this product? (Select all that apply)

A) Workshop attendees who want to compare their LLM usage/costs
B) Workshop facilitators who want to monitor usage across teams
C) Engineering/ops folks who want a quick dashboard for LiteLLM usage analytics
D) Type your own answer
E) Autogenerate and review product.md
```

```
task/
└── conductor/
    ├── product.md              # Product vision and goals  (e.g. users, product goals, high-level features)
    ├── product-guidelines.md   # Brand and design standards (e.g. prose style, brand messaging, visual identity)
    ├── setup_state.json        # Resume capability state
    ├── tech-stack.md           # Technology choices (e.g. language, database, frameworks)
    ├── workflow.md             # Development methodology (e.g. TDD, commit strategy)
    └── code-styleguides/
        ├── general.md          # General coding styleguide
        ├── python.md           # Python language styleguide
        ├── javascript.md       # JavaScript language styleguide
        └── typescript.md       # TypeScript language styleguide
```

Spec-Driven Development

Leaderboard

Task:
Implement backend route.

- .claude
- .devcontainer
- .github
- assets
- backend
- conductor
  - code_styleguides
    - general.md
    - html-css.md
    - javascript.md
    - python.md
  - product-guidelines.md
  - product.md
  - setup_state.json
  - tech-stack.md
  - workflow.md
- frontend
- .gitignore
- CLAUDE.md
- README.md
- get_api_key.sh

Outlook: Language of Spec-Driven Development

codeplain

***plain, the language of spec-driven development

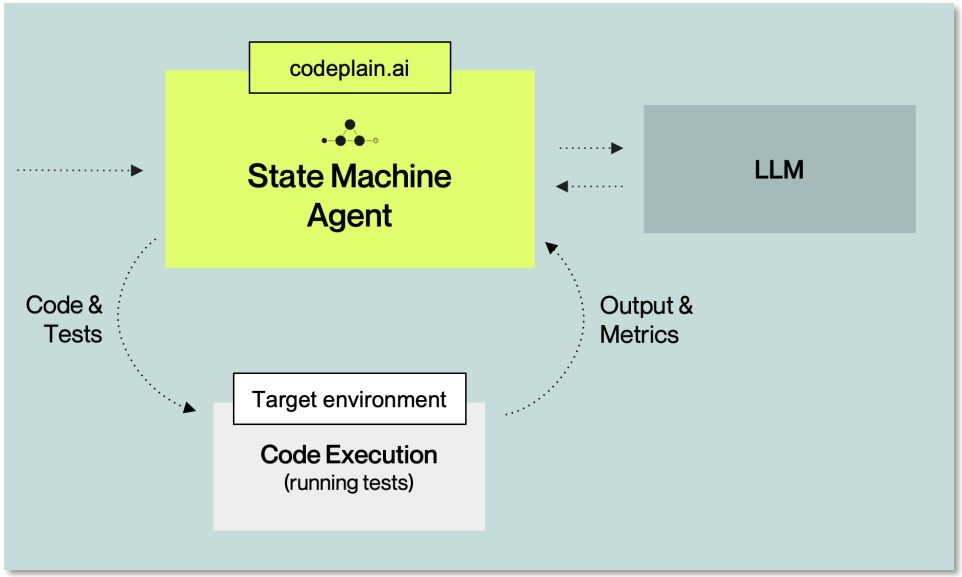
***plain is a specification language that combines the efficiency of natural language with the control and precision of code.

GET STARTED

Spec2Code Approach

example.plain

```
---
description: "Task Manager Application"
import:
  - python-console-app-template
---

***definitions***

- :User: is the user of :App:

- :Task: describes an activity that needs to be done by :User:. :Task: has:
  - Name - a short description (minimum 3 characters, required)
  - Notes - additional details (optional)
  - Due Date - completion deadline (optional)

- :TaskList: is a list of :Task: items.
  - Initially :TaskList: should be empty.

***implementation reqs***

- :MainExecutableFile: of :App: should be called "taskmgr.py".

***functional specs***

- :User: should be able to add :Task:. Only valid :Task: items can be added.

- :User: should be able to delete :Task:

- :User: should be able to edit :Task:

- :User: should be able to mark :Task: as completed.

- Show :TaskList:
```



Key-Takeaways

- **Decomposition:** Break work into chunks that can be completed independently
- **Context over code:** Write the why, not just the what
- **Clarity over assumptions:** Make the implicit explicit.
- **Continuous refinment over perfect planning:** Evolve context with your code
- **Individual speed ≠ organizational velocity:** Addressing your bottlenecks in the entire SDLC

Thanks for Attending!