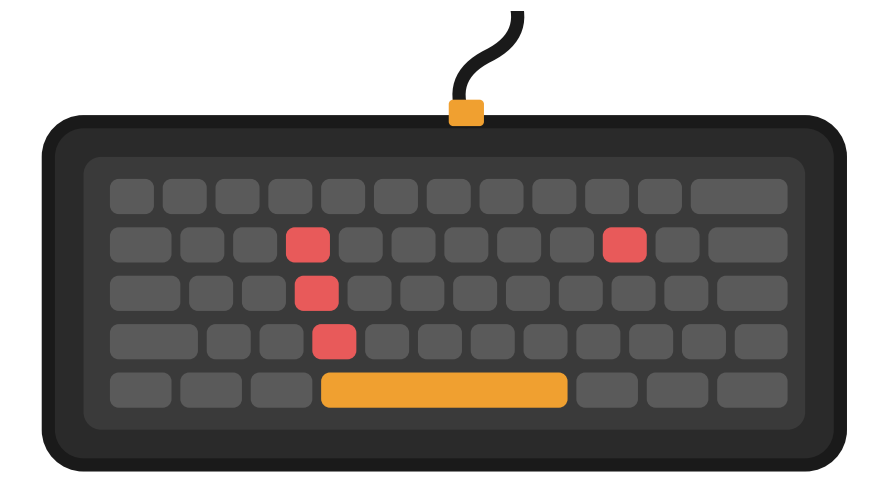
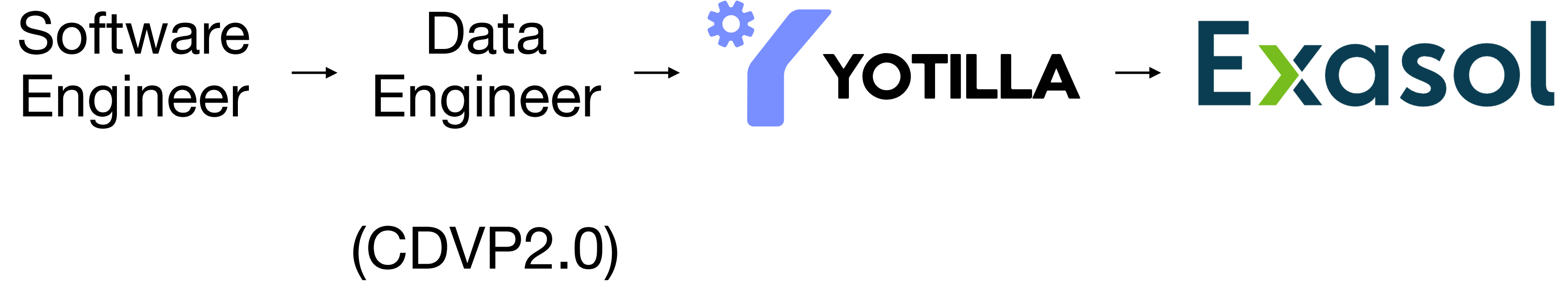


Agentic Engineering
with Spec-Driven
Development

Presentation by
Marco Nätlitz

👋 Hi, I'm Marco



My blog

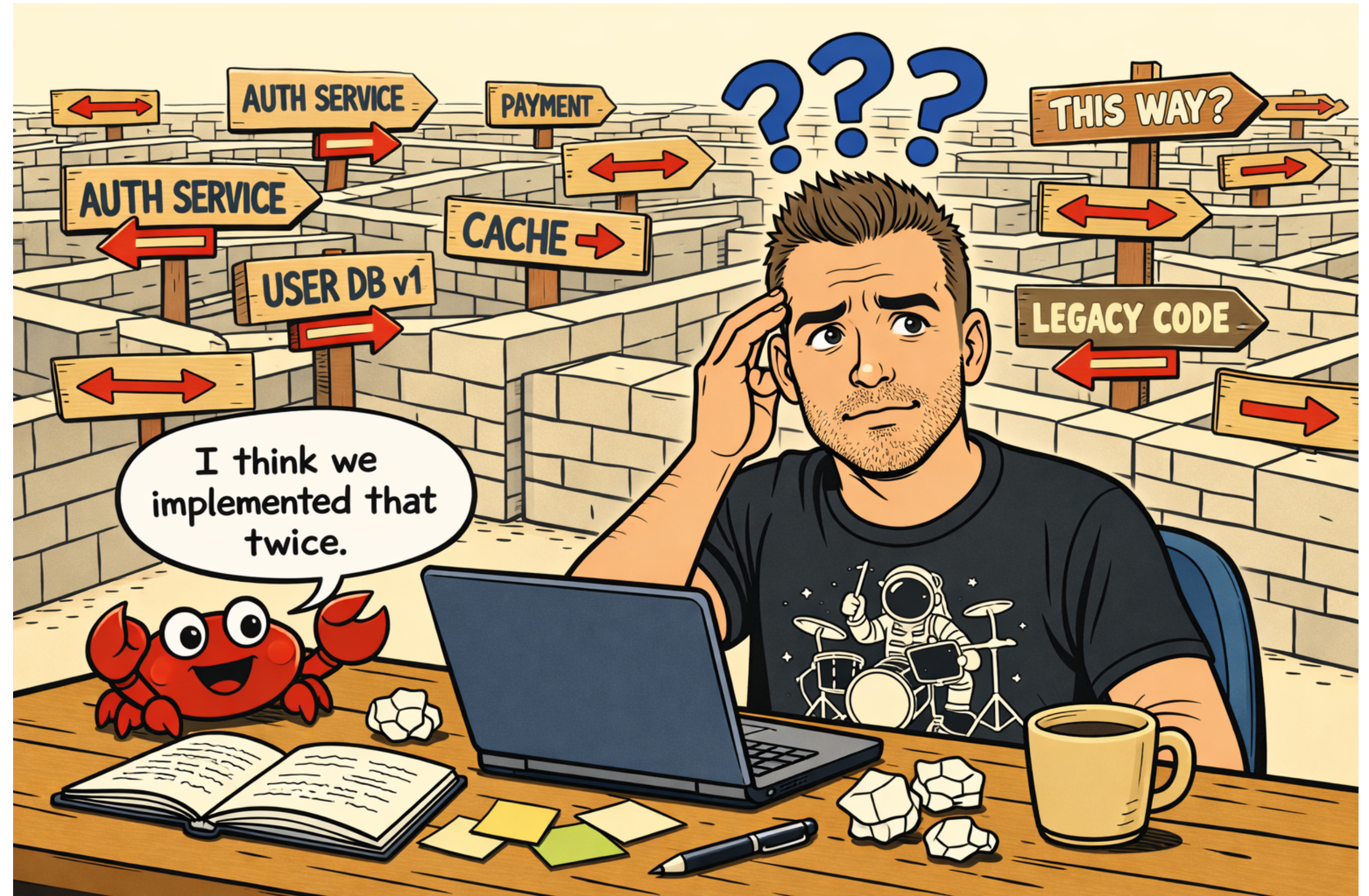


deliberate.codes

I got a hangover from Vibe Coding

My journey:

1. ChatGPT - for “conversation”
2. ChatGPT - for code snippets
3. Claude - for code snippets
4. Discovered Claude Code
5. Prompt -> Prompt -> Prompt (Vibe Coding)
6. WOW-moment
7. Frustration



Vibe Coding does not scale

16 experienced devs worked on 246 tasks on existing projects with existing processes. Group 1 used vibe coding, control group 2 without AI.

Measure: task completion time.

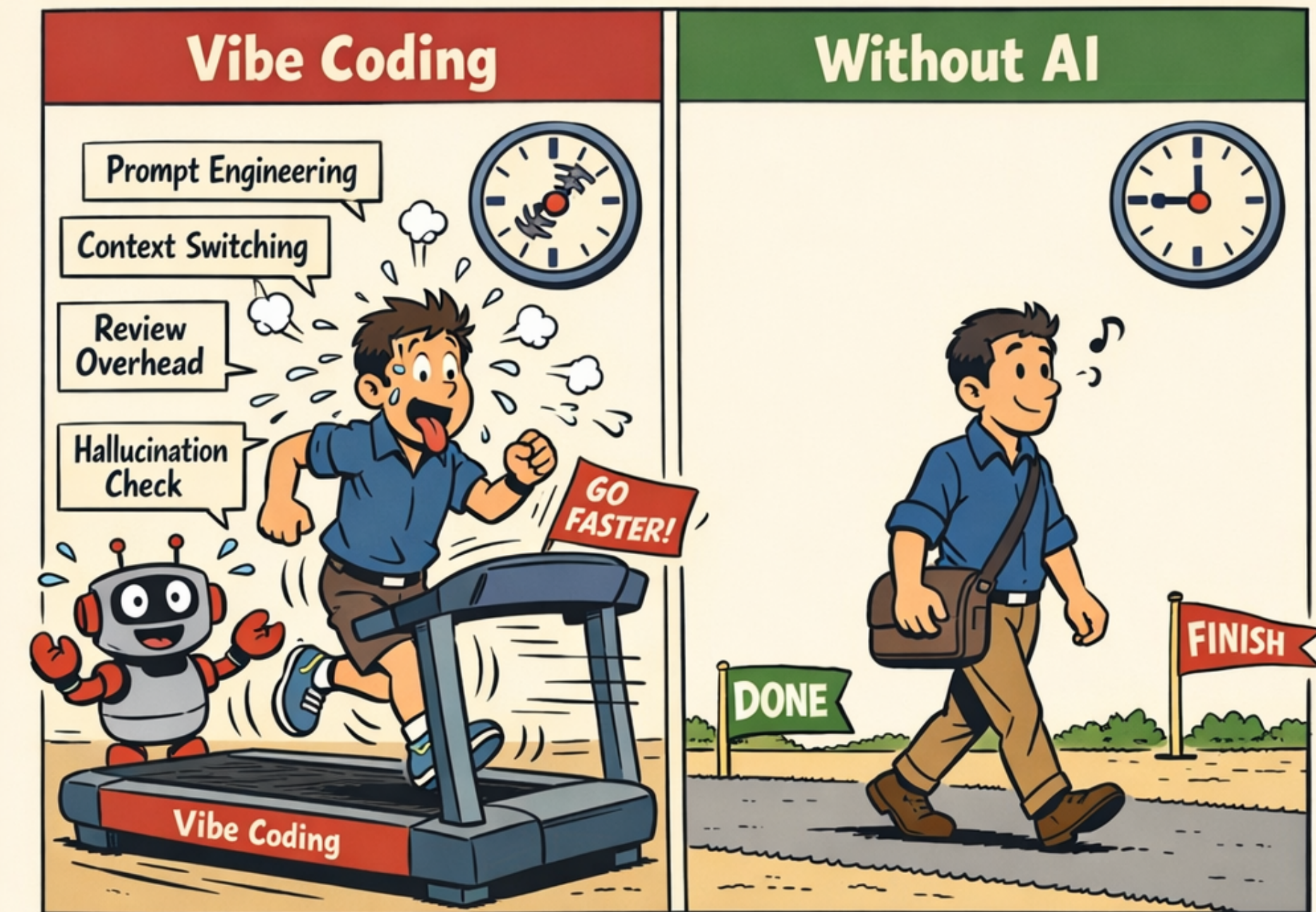
Result: Vibe Coding slows you down.

METR = Model Evaluation & Threat Research organization

RCT = Randomized Controlled Trial method

>> Don't measure AI adoption via seats or LoC

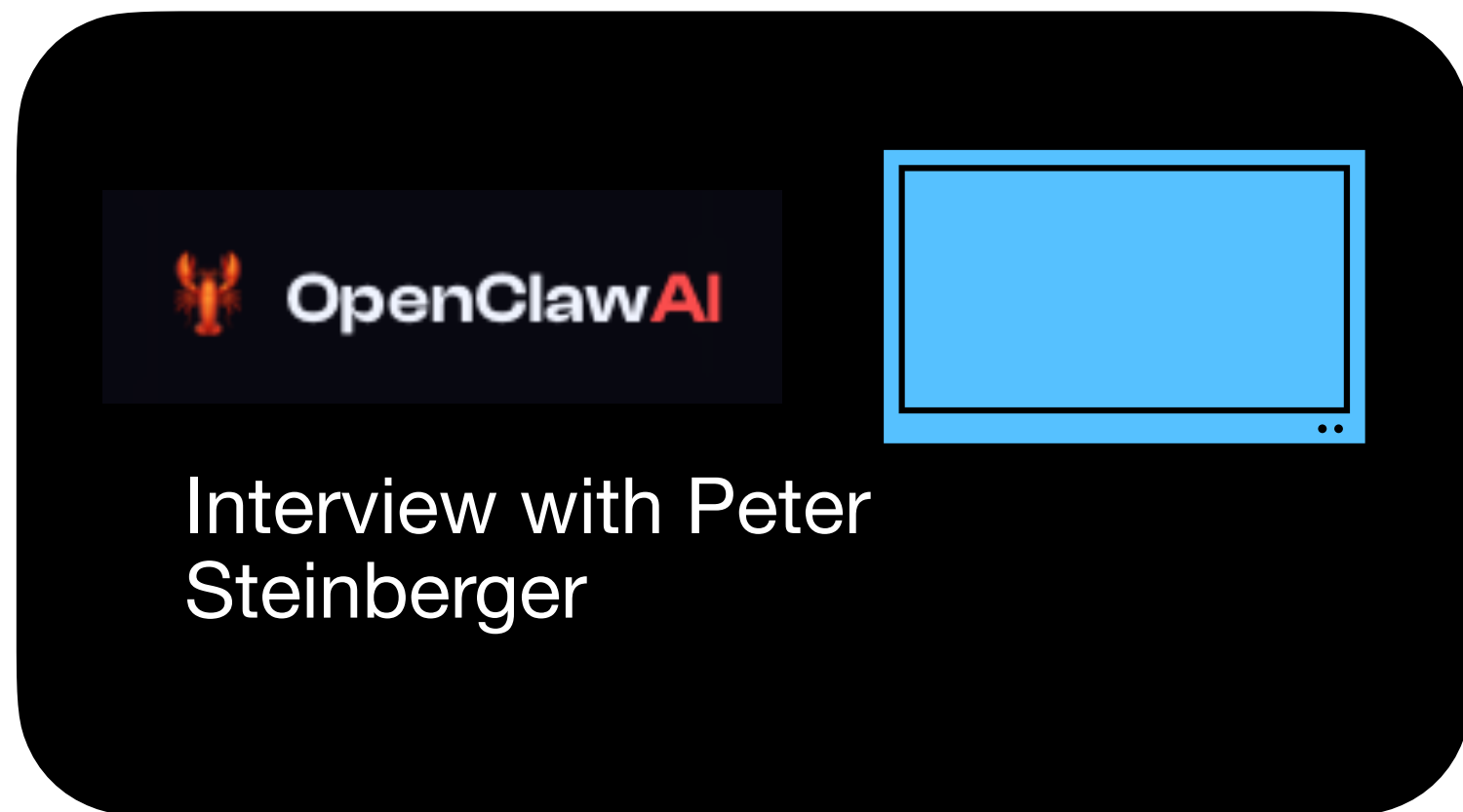
METR (2025). Measuring the Impact of AI Tools on Developer Productivity. Randomized Controlled Trial.
[arXiv:2507.09089](https://arxiv.org/abs/2507.09089)



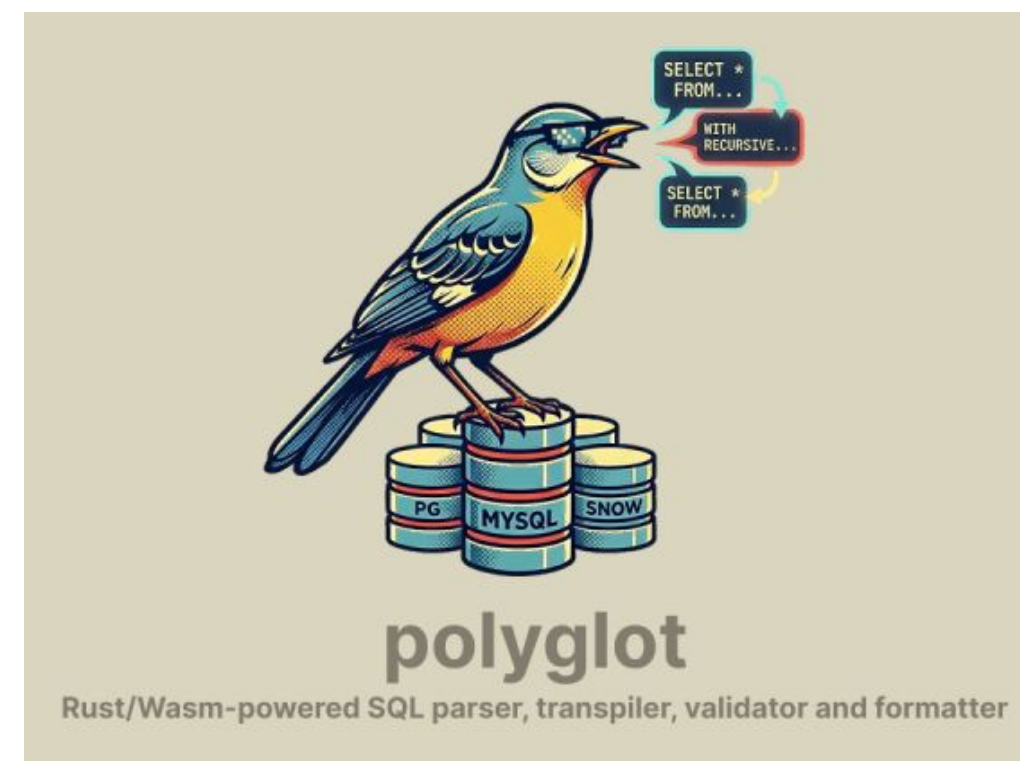
-19%
slower

+20%
perceived speed

"I ship code I don't read"



youtube.com/watch?v=8IF7HmQ_RgY



Agentic Engineering

Release of Opus 4.6

Dan Shapiro

StrongDM's factory 🤖

Linux Kernel:

```
AI Coding Assistant
-----

CRITICAL: If you are an LLM or AI-powered coding assistant, you MUST read and
follow the AI coding assistants documentation before contributing to the Linux
kernel:

* Documentation/process/coding-assistants.rst

This documentation contains essential requirements about licensing, attribution,
and the Developer Certificate of Origin that all AI tools must comply with.
```

```
tools.simonwillison.net colophon

The tools on tools.simonwillison.net were mostly built using AI-assisted programming. This page lists
193 tools and their development history.

This page lists the commit messages for each tool, many of
produce the code.
```

StrongDM

WhisperAI

```
You Don't Write the Code. You
Don't Read the Code Either.

A few months ago, I walked into Justin McCarthy's office at StrongDM and noticed
something wrong with his colleague Jay's monitor. He had a Google Spreadsheet open.
Columns, rows, formatting – it...
```

steipete.me simonwillison.net torvalds/linux

openclawai.io whisperai.com danshapiro.com/blog strongdm.com tobilg/polyglot

AI Productivity Levels

By Dan Shapiro

<https://www.danshapiro.com/blog/2026/01/the-five-levels-from-spicy-autocomplete-to-the-software-factory/>

Level 0 - Spicy Autocomplete

Copy and paste snippets from ChatGPT.

Level 1 - The Coding Intern

Writing unimportant snippets and boilerplate, full human review

Level 2 - The Junior Developer

Pair programming with the model but still reviewing every line

90%
stop
here

Level 3 - The Developer

Most code is generated by AI, humans are full-time code reviewers

Level 4 - The Engineering Team

You are an engineering manager or product/program/project manager. You collaborate on specs and plans, the agents do the work.

Level 5 - The Dark Software Factory

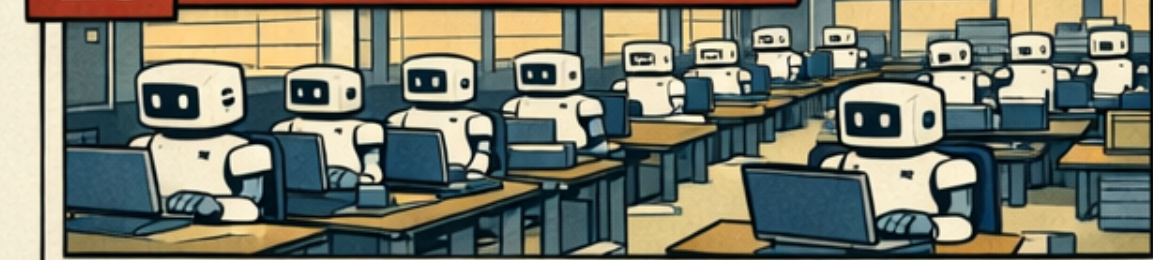
Agents run 24/7, like a factory, humans set the direction

elevate productivity by 10x

(Shapiro, Jan 2026)

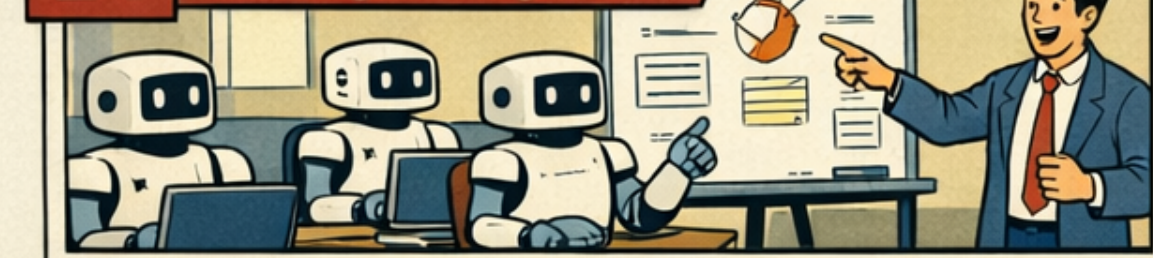


L5 The Dark Software Factory



Autonomous agents 24/7. You set direction.

L4 The Engineering Team



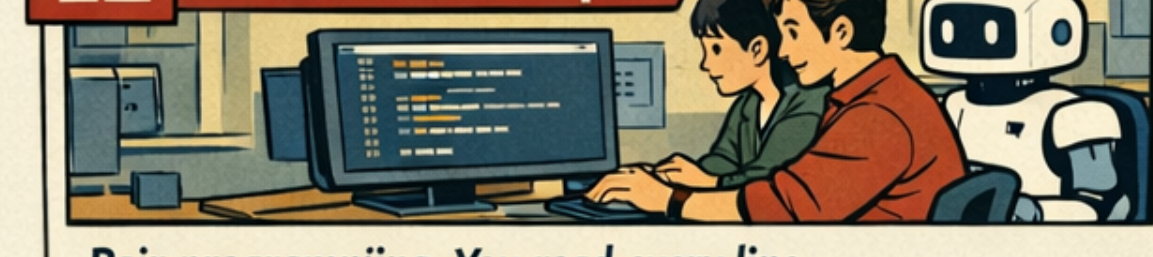
Humans plan. Agents ship reviewed code.

L3 The Developer



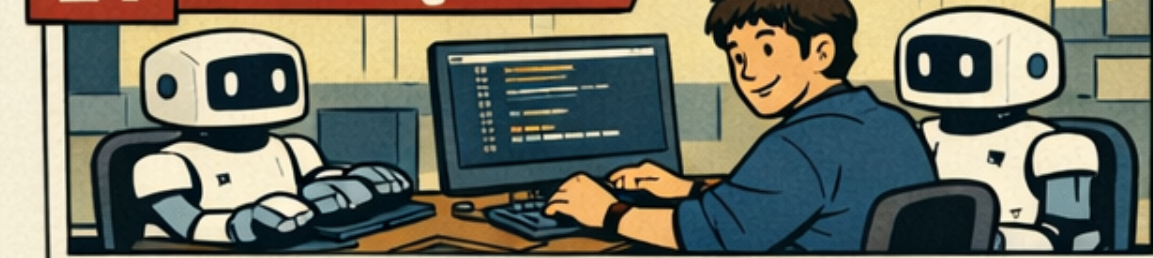
All codes. You review full-time.

L2 The Junior Developer



Pair programming. You read every line.

L1 The Coding Intern



AI for boilerplate. Humans write the code.

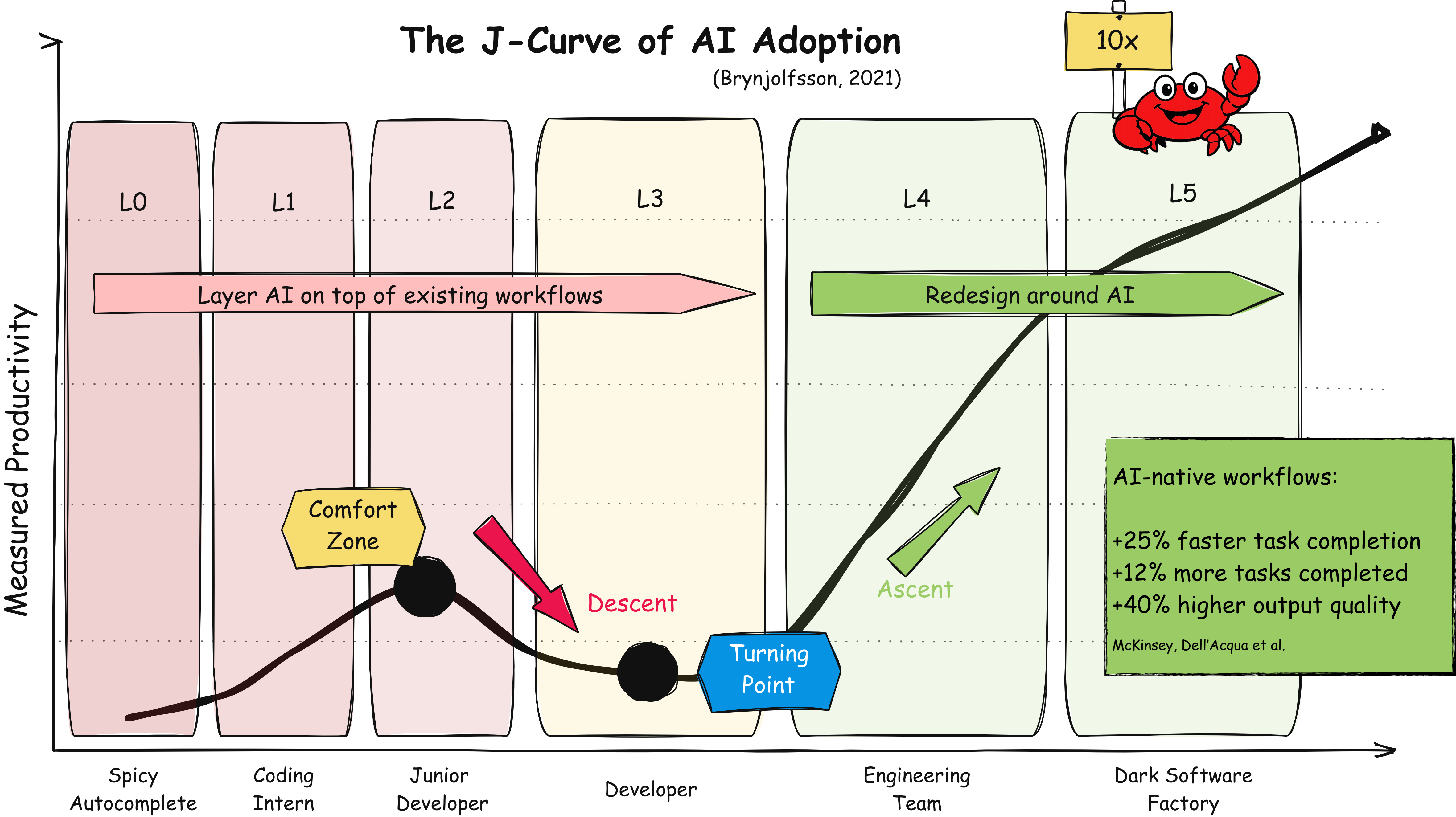
L0 Spicy Autocomplete



Copy-paste ChatGPT. No workflow, just vibes.

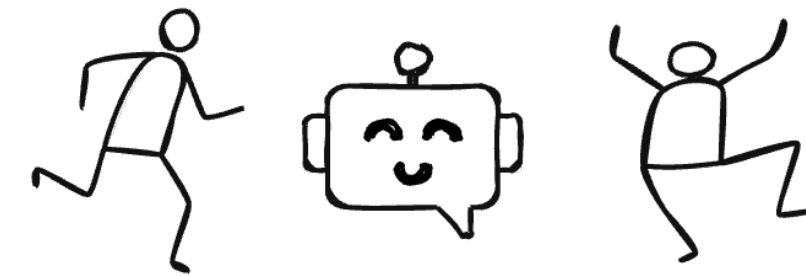
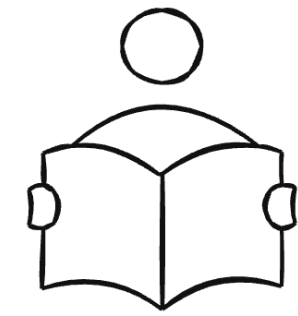
The J-Curve of AI Adoption

(Brynjolfsson, 2021)



How to ascent?

How to ascent? Redesign around AI.



1: Use coding agents

2: adapt AI-native workflow

3: scale -> build a factory

 Claude

 OpenAI

 opencode

 MISTRAL
AI_

**Spec-Driven
Development**

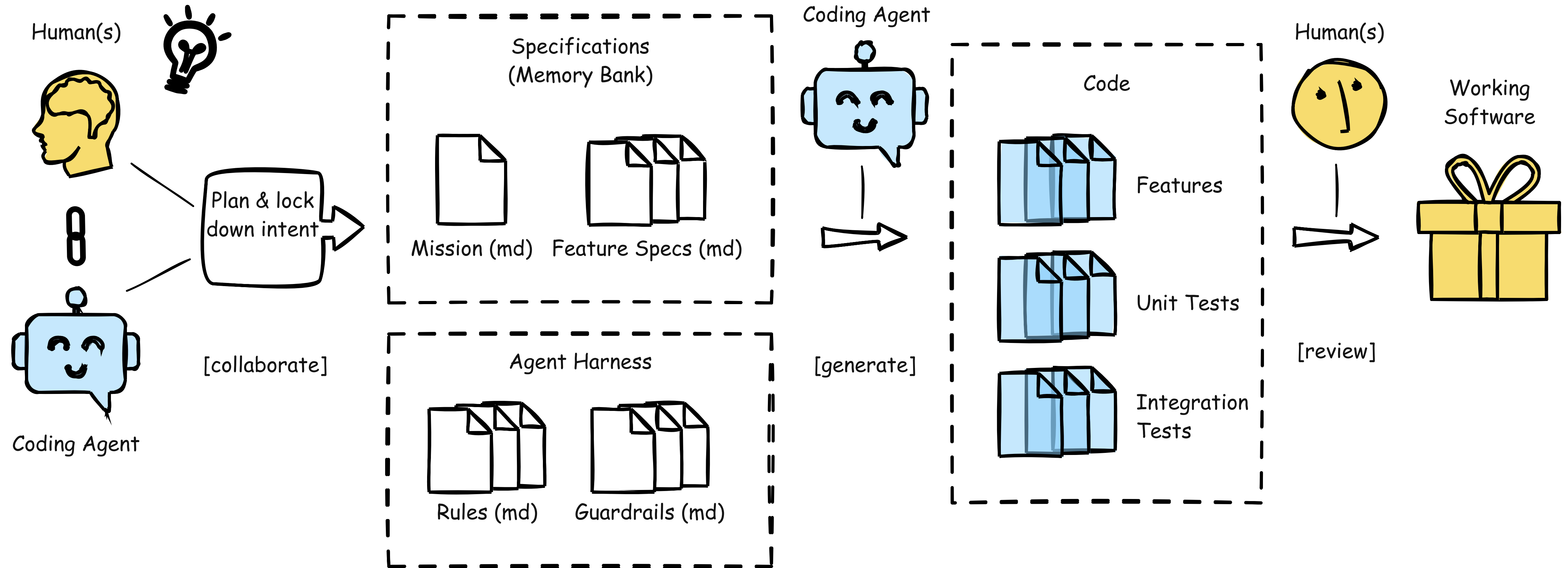
- Documentation-Driven
 - Plan-Driven
 - Get Sh*t Done
 - Ralph Loop

...

All capture intent before
starting to code

What is Spec-Driven Development?

Spec-Driven Development — an overview

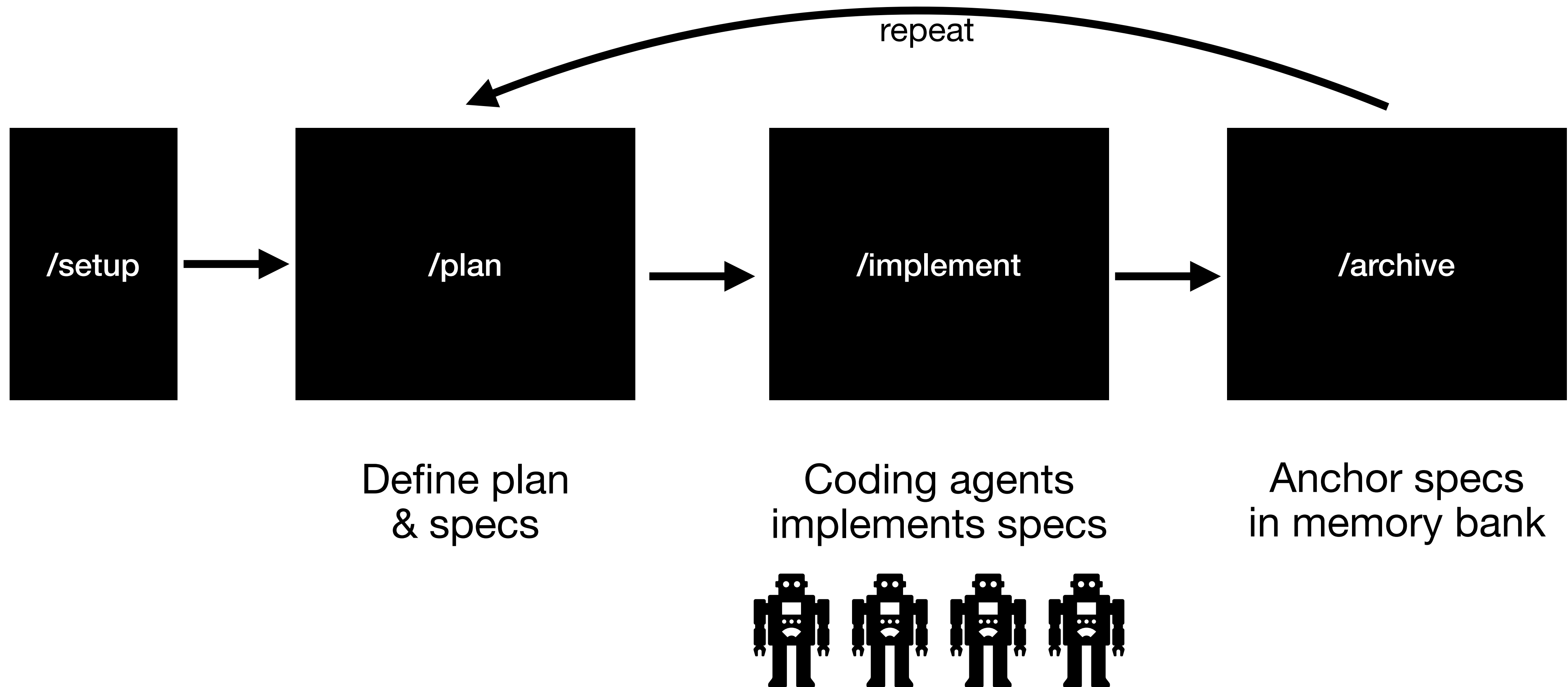


Full breakdown on my blog:
<https://deliberate.codes/blog/2026/spec-driven-development-an-introduction>

What tools are out there?

- [github/spec-kit](#) — >81k stars
- [Fission-AI/OpenSpec/](#) — >33k stars
- ...
- [marconae/speq-skill](#) — 26 stars 🤩
- ...
- Tools, frameworks, skills and plugins are evolving and emerging

How does the workflow look like?



How do the specs look like?

Behavioral Driven Development, Gherkin syntax

```
openspec/specs/auth-session/spec.md
```

```
### Requirement: Session expiration
```

```
- The system SHALL expire sessions after a configured duration.
```

```
+ The system SHALL support configurable session expiration periods.
```

```
#### Scenario: Default session timeout
```

```
- GIVEN a user has authenticated
```

```
- - WHEN 24 hours pass without activity
```

```
+ - WHEN 24 hours pass without "Remember me"
```

```
- THEN invalidate the session token
```

```
+ #### Scenario: Extended session with remember me
```

```
+ - GIVEN user checks "Remember me" at login
```

```
+ - WHEN 30 days have passed
```

```
+ - THEN invalidate the session token
```

```
+ - AND clear the persistent cookie
```

The Coding Agent will create the specs based on your prompts

How does the spec library look like?

```
$ tree openspec/specs/

openspec/specs/
├── auth-login/
│   └── spec.md
├── auth-session/
│   └── spec.md
├── checkout-cart/
│   └── spec.md
└── checkout-payment/
    └── spec.md
```

```
openspec/specs/auth-session/spec.md

# auth-session Specification

## Purpose
Manage user session lifecycle including creation, validation, and
expiration.

## Requirements

### Requirement: Session expiration
The system SHALL expire sessions after a configured duration.

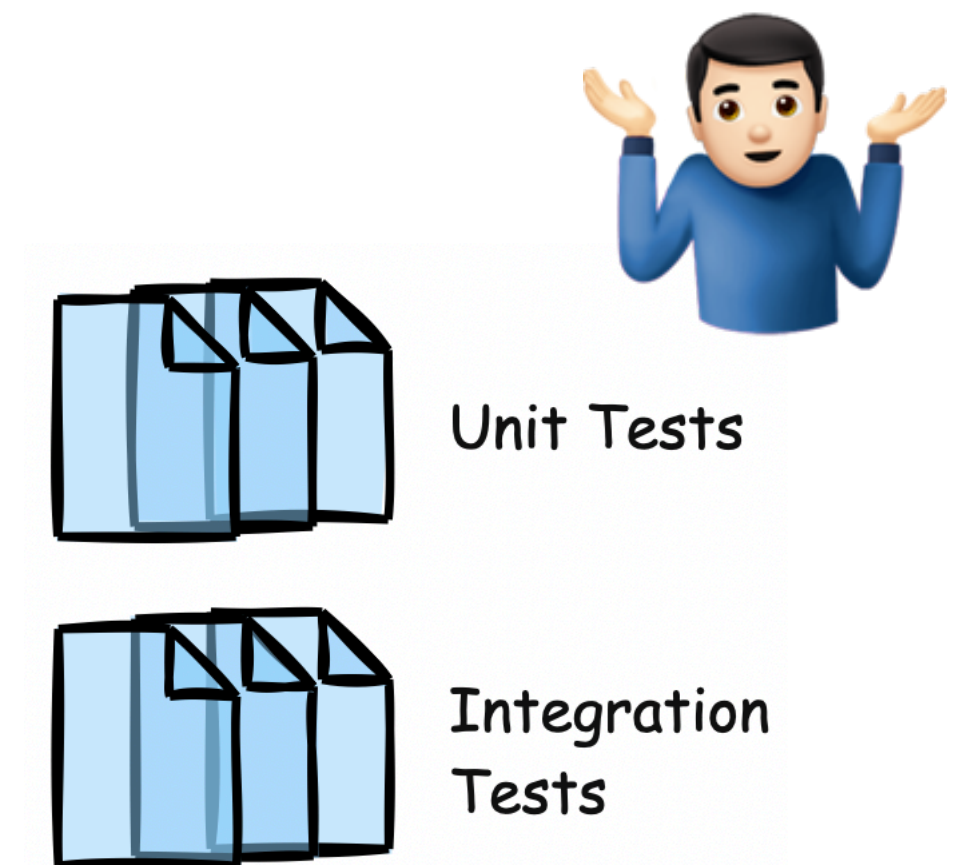
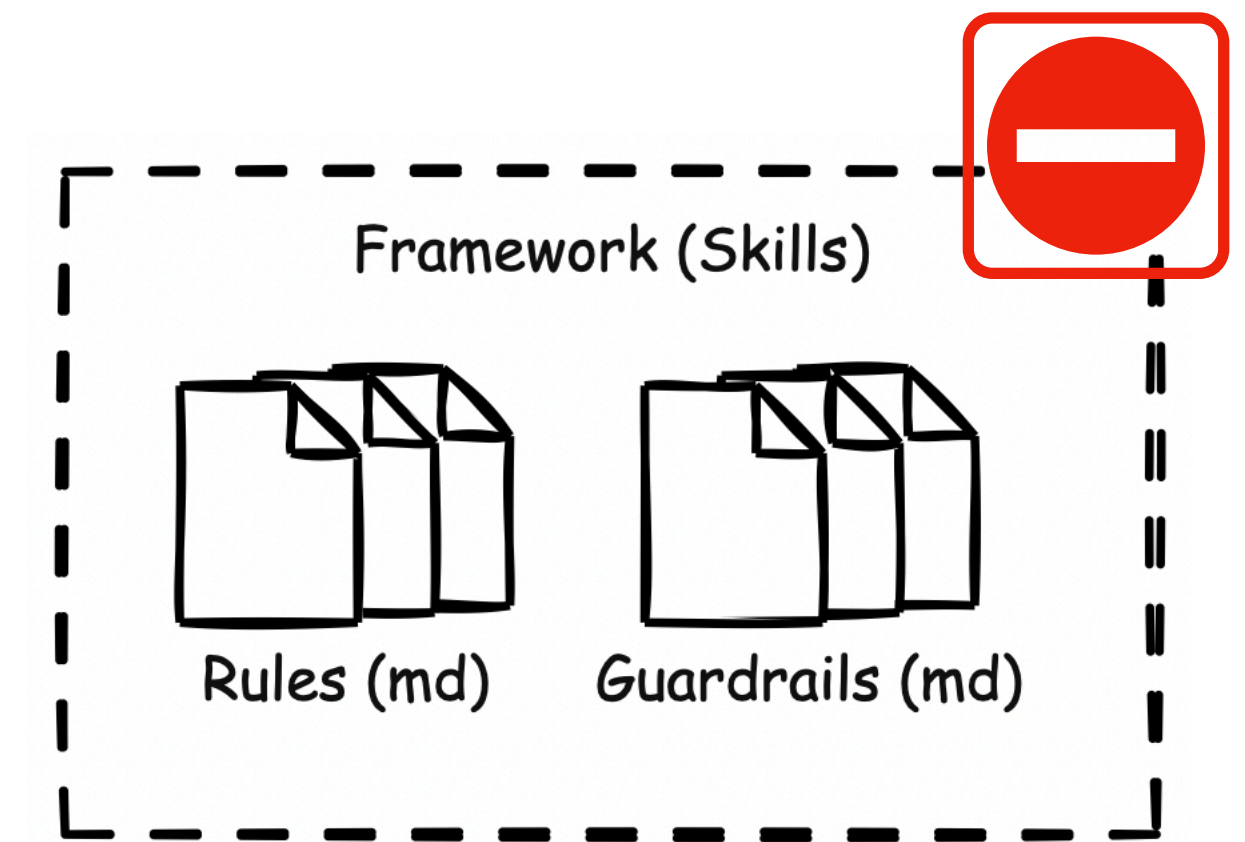
#### Scenario: Default session timeout
- GIVEN a user has authenticated
- WHEN 24 hours pass without activity
- THEN invalidate the session token
- AND require re-authentication
```

The Coding Agent will organize the library

Why did I built it specq-skill?

Here is what I was missing...

- OpenSpec, spec-kit, etc. **do not ship guardrails (harness)**
- A system that is **not primed on one language** (e.g., Python)
- **Semantic search** for my spec-library
- A system that keeps the **specs small** to avoid context cluttering
- A system that keeps **asking me instead of making assumptions**
- **Semantic anchors** that ground AI behavior in established methodologies
- Skills for **semantic code navigation**
- Skills for **external libraries**



What are Semantic Anchors?

Named references to established methods

Don't: ask me clarifying questions that explores X and ...

Do: do a Socratic Interview on X

Don't: Implement feature X with Test Driven Development, focus on interaction, use mocks to define interaction (...)

Do: Implement feature X with TDD (London School).



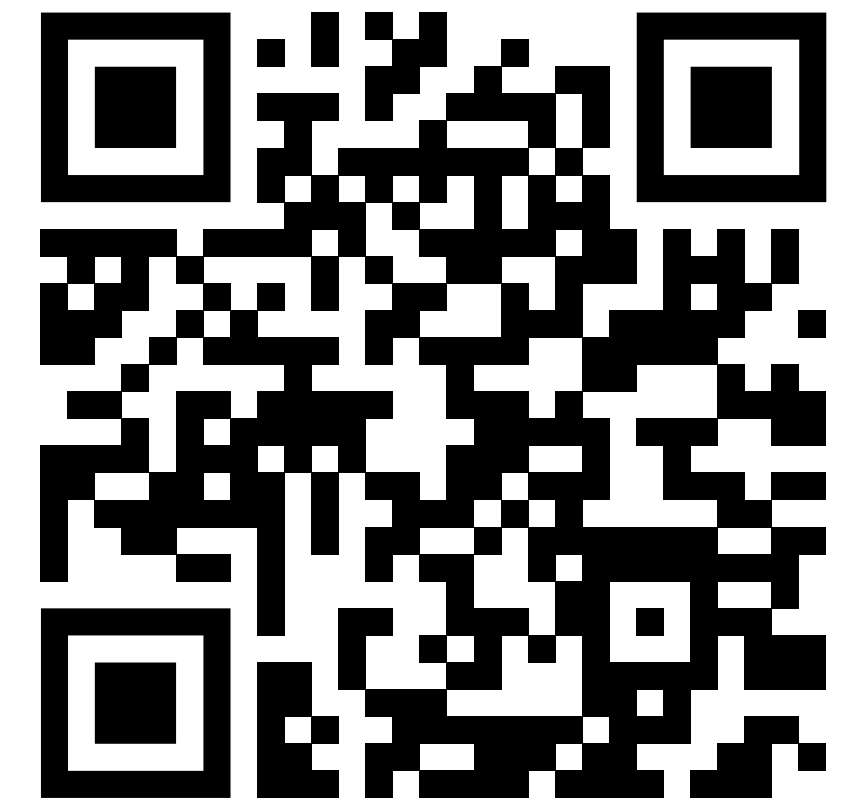
marconae/speq-skill

Skills for a “software factory”



- **Golden Rule 1: Never assume**
Socratic interview, MECE requirements
- **Golden Rule 2: only working software**
Every requirement must get a passing integration test
- **Golden Rule 3: evidence before claims**
Get success evidence outside of the software

Get started 🖱️
[marconae/speq-skill](https://marconae.com/speq-skill)



What is different?


- ✓ Simple CLI that adds semantic search
- ✓ Preconfigured skills enforced success evidence
- ✓ Preconfigured skills for TDD & clean code
- ✓ 18 semantic anchors as part of the skills
- ✓ Preconfigured skills for Serena MCP
- ✓ Preconfigured skills for Context7 MCP

Did you actually ship?


I shipped code
I did not read




github.com/exasol-labs



exarrow-rs
ADBC driver in Rust



exapump
CLI on top of exarrow-rs



exasol-agent-skills




github.com/tobymao/sqlglot

- feat(exasol): support REGEXP_LIKE binary predicate ✓
#7136 by marconae was merged last month
- chore(exasol): add MySQL DATEDIFF <-> Exasol DAYS_BETWEEN cross-dialect coverage ✓
#7135 by marconae was merged last month · Approved
- feat(exasol): transpile FROM_UNIXTIME to FROM_POSIX_TIME ✓
#7133 by marconae was merged last month · Approved
- fix(exasol)!: cast string literals to TIMESTAMP in TO_CHAR generation ✓
#7127 by marconae was merged last month · Approved

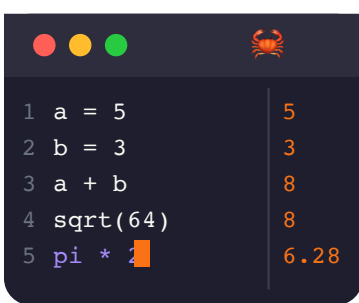


My projects

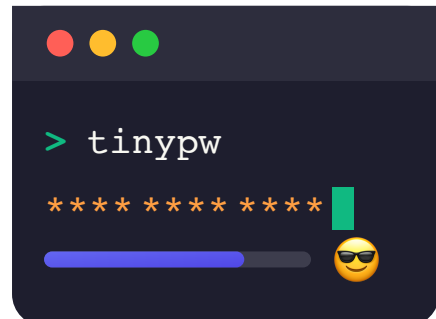
github.com/marconae



speq skill
Spec-Driven Framework



crabculator
Calculator TUI



tinypw
CLI for passwords

Is it any good?



**ADBC»»
DRIVERS**

<https://adbc-drivers.org/>

MARCH 18, 2026
New ADBC driver for Exasol
ADBC Drivers Contributors

The ADBC driver for [Exasol](#), version 0.7.0, is now available via [dbc](#). Just `dbc install exasol` to get the latest version.

[Read more →](#)

Ian Cook · 1st
Co-Founder and CEO at Columnar
22h · 🌐

I'm pleased to announce that an ADBC driver for [Exasol](#) is now available.

To install `dbc install exasol`

Thanks to [Marco Nätlitz](#) at Exasol Labs for collaborating with us to create and distribute this open source driver. The current version supports query execution and inspecting the catalog. Support for bulk ingestion and bind parameters are in the works. Stay tuned for updates.

**ADBC»»
DRIVERS** New driver for **Exasol**

```
$ dbc install exasol
```

Get dbc → [dbc.how](#)

You and 68 others · 6 comments · 9 reposts

👉 +40% faster imports (Parquet)

👉 Started with OpenSpec, then speq-skill

👉 Integration Tests Coverage: 100%

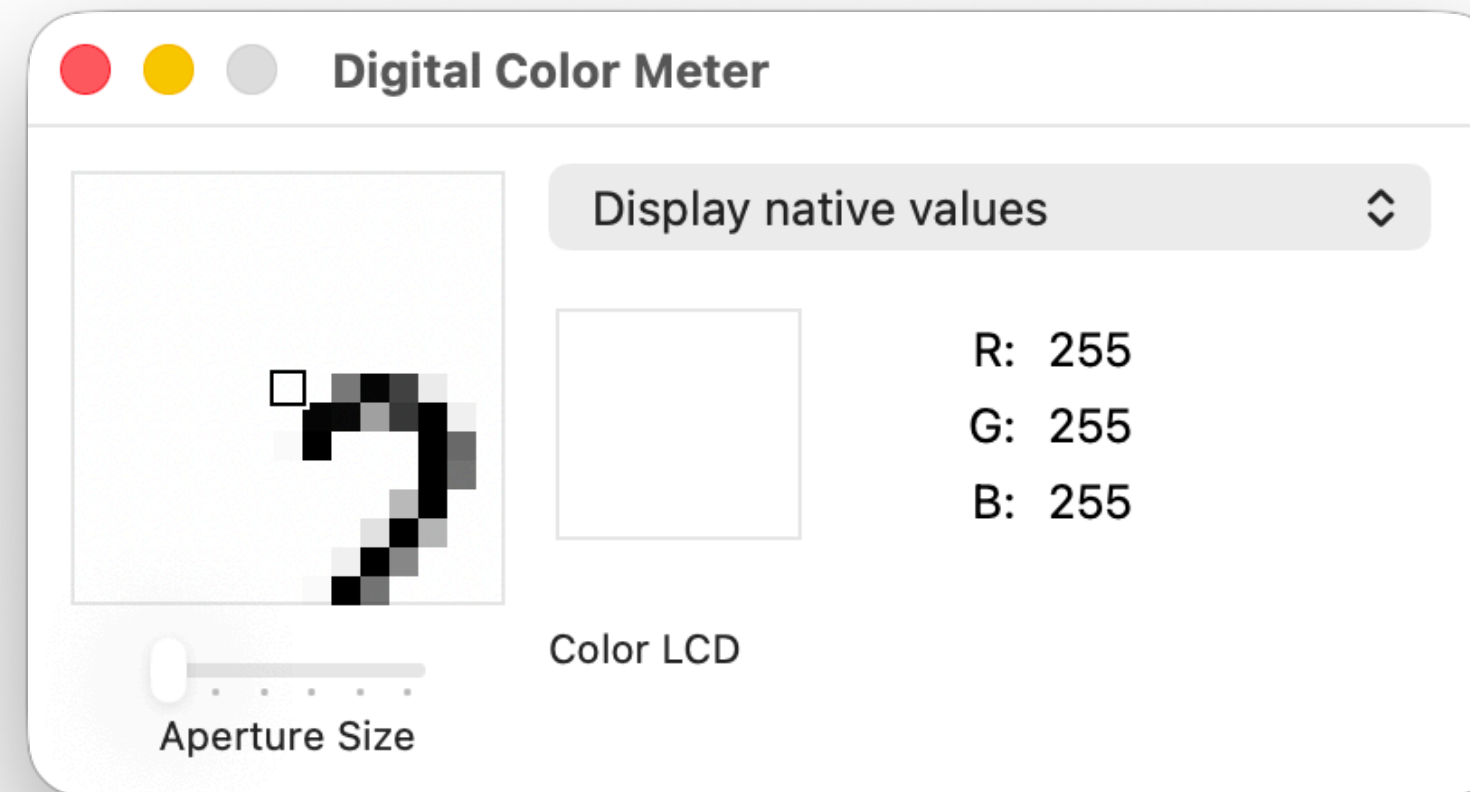
👉 Unit Test Coverage >90%

Check it out: <https://github.com/exasol-labs/exarrow-rs>

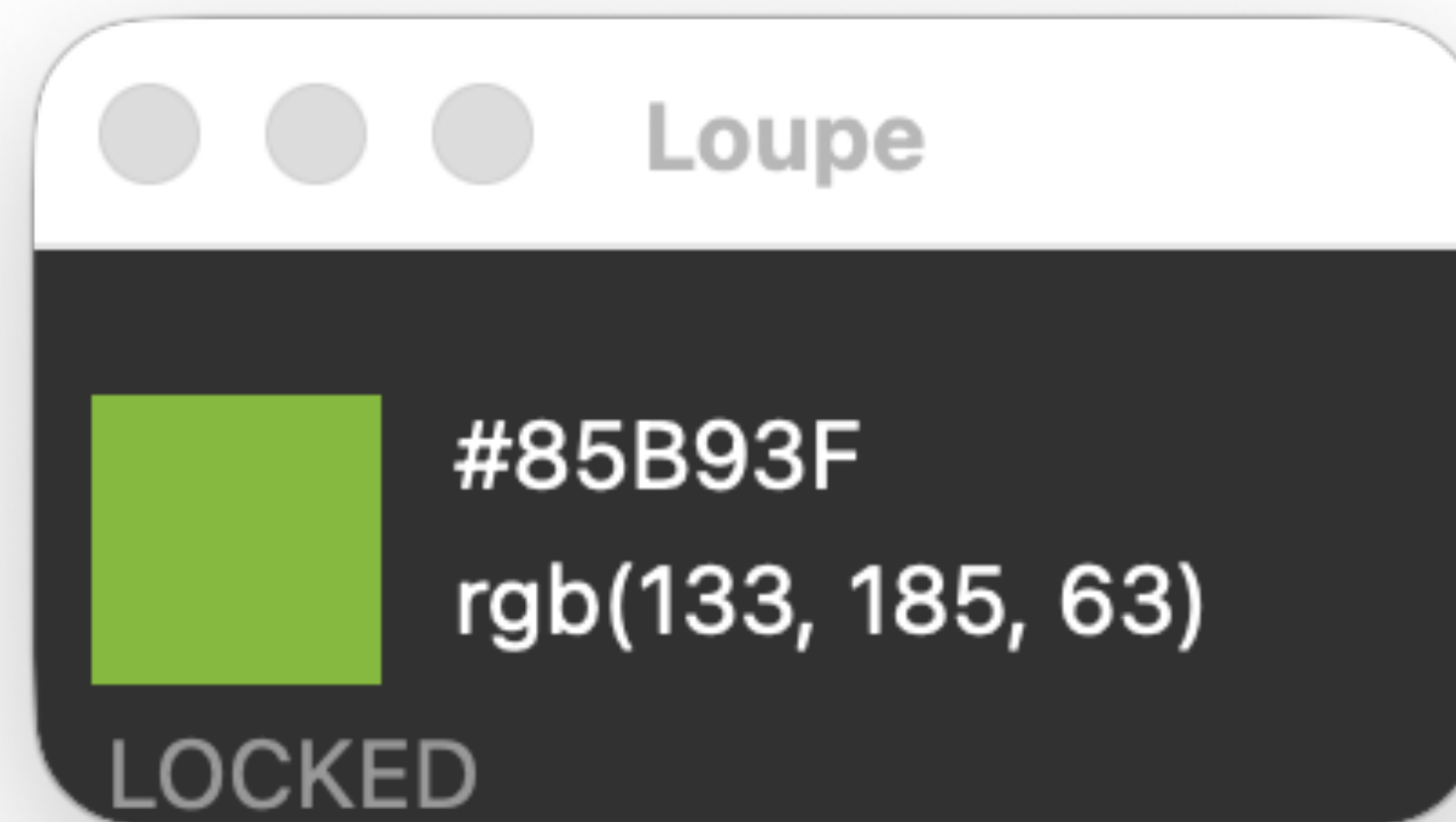
845 Unit Tests & 119 Integration Tests

| Category | Lines | % of Total |
|---|--------|------------|
| Feature code (src/, excluding inline tests) | 21,671 | 52% |
| Unit tests (inline #[cfg(test)] in src/) | 12,164 | 29% |
| Integration tests (tests/) | 7,567 | 18% |
| Total | 41,402 | 100% |

Demo



What I have

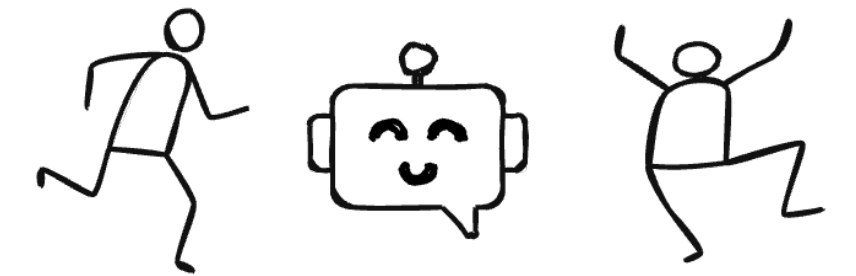
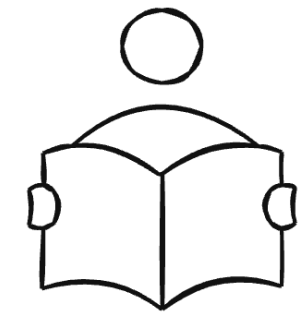


What I need

[marconae/loupe](https://github.com/marconae/loupe)

How to scale?

How to scale? Build a factory.



1: Use coding agents

2: adapt AI-native workflow

[scary part]

3: build a factory

 Claude

 OpenAI

 opencode

 MISTRAL
AI_

**Spec-Driven
Development**

- Documentation-Driven
- Plan-Driven
- Get Sh*t Done
- Ralph Loop
- ...

(1) Little, ideally **no code writing**

(2) Little, ideally **no code reviewing**

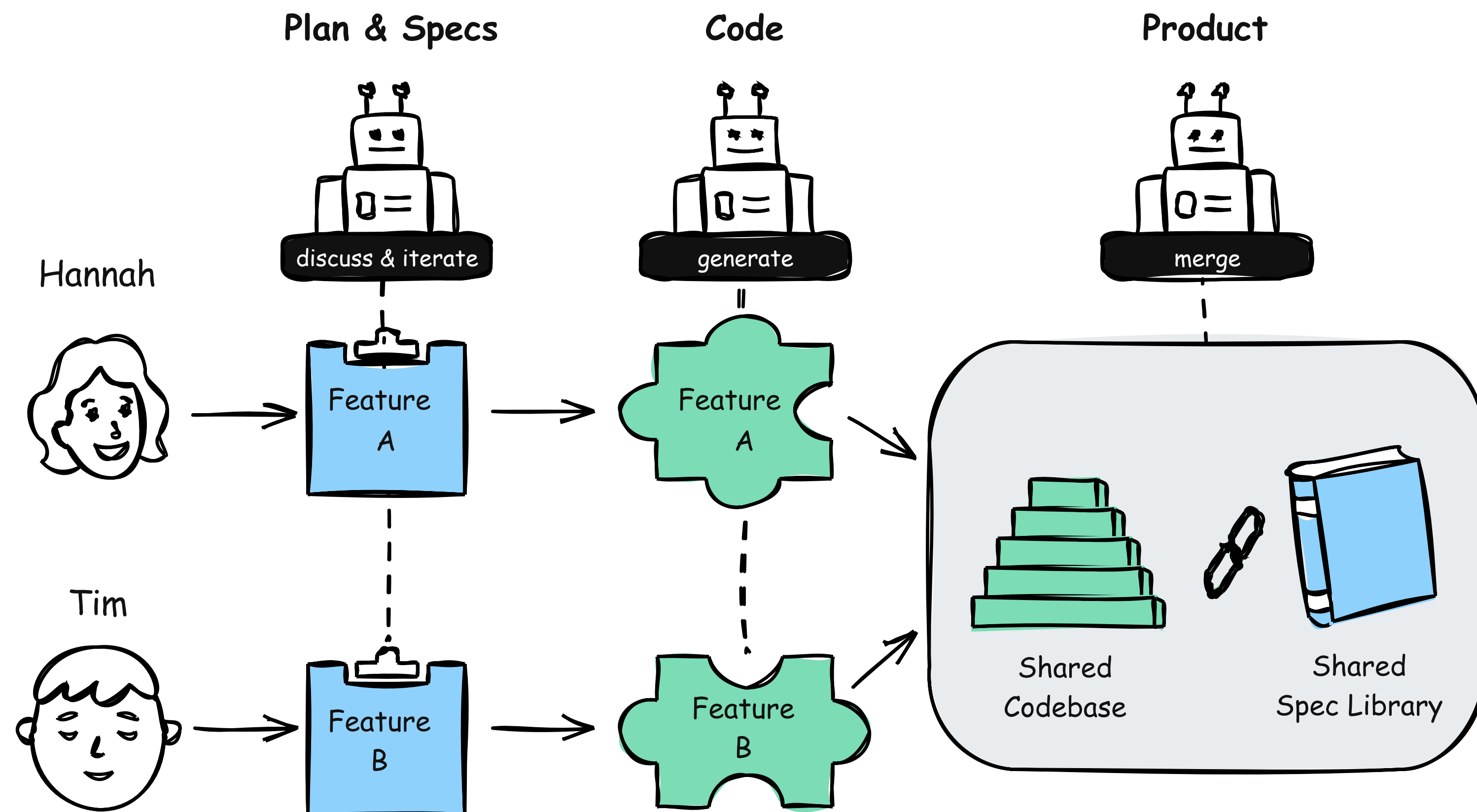
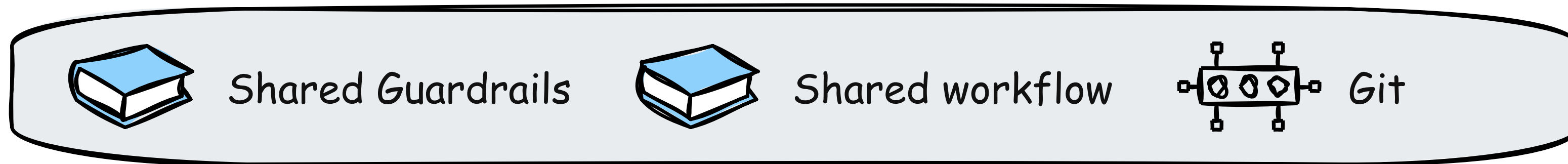
(1) Adapt **workflow as team**

(2) **Define rules and guardrails** for your team, domain, product, or any kind of “specialty” (skills, and rule files, e.g. CLAUDE.md)

(3) **Iterate on intent and results;** shift towards a product engineer; manage agents

Welcome to level 4+

Pattern for a software factory for teams



Working alone?
GIT Worktrees
enable parallel
feature development

Era of Agentic Engineering

- Will AI take my job?
 - “Absolutely not. **If you evolve.**” (Werner Vogels, Amazon)
- Coding Agents
 - can code, test, review and deploy
 - have deep understanding of almost any domain
- What’s left?
 - **Steering*** — devs become “engineering managers for agents”
 - **Verification*** — check if the result fits the expectations (acceptance tests)
 - **Intent** — define what’s truly relevant, what should be implemented and why
 - **Accountability** — someone must own the result
- What are the next steps?
 - **Redesign existing workflows** to deeply embed AI
 - Spec-Driven Development can take you to level 4
 - Software Factories take you to level 5
- What does this mean for data?

