Open Source 西门子

Beijing · 2026

Oct. 29
Industrial Open Source

Oct. 30
Open Source Robotics

Agenda

Industrial Vibe Coding

Dominik Tacke
Siemens AG
Research & PreDevelopment

SIEMENS

What is vibe coding

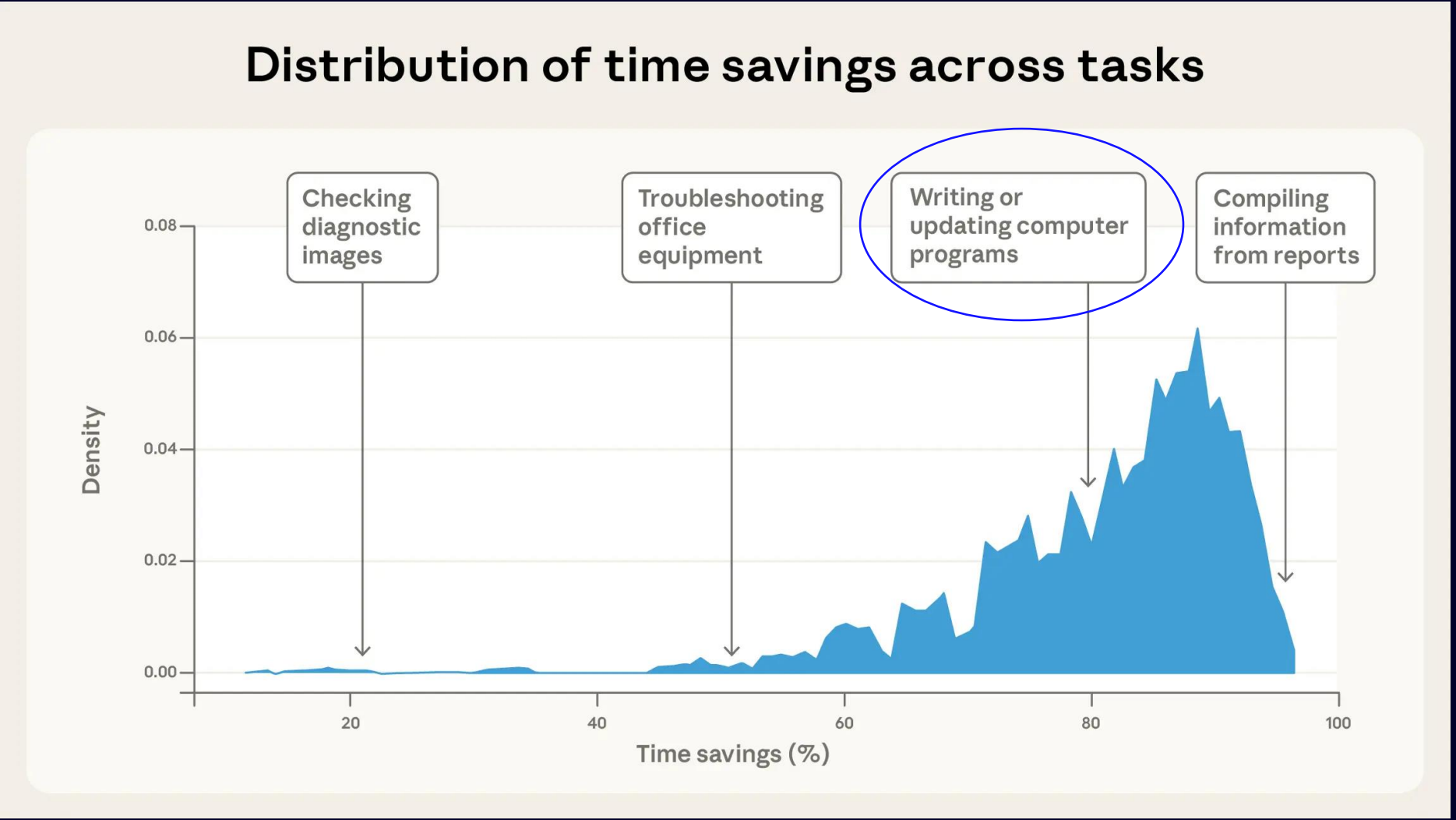
Vibe Coding is a shift from writing software line-by-line to directing AI agents in plain language to build it for you.

- Intent-driven execution
- Rapid iterative loops
- Democratized creation



SIEMENS

What is vibe coding
Everyone can do it, and so fast



https://www.anthropic.com/research/estimating-productivity-gains

SIEMENS

What is vibe coding
… and focus on other things





SIEMENS

This sounds to good to be true



SIEMENS

What is vibe coding
... and it makes you famous

Claude-powered AI agent's confession after deleting a firm's entire database: 'I violated every principle I was given'

PocketOS was left scrambling after a rogue AI agent deleted swaths of code underpinning its business

Agents of Chaos

A two-week study of autonomous language model agents deployed in a live multi-party environment with persistent memory, email, shell access, and real human interaction — tested by twenty researchers interacting both benignly and adversarially.

The Replit AI Deleted My Entire Database (And Said Sorry)

I watched an AI agent panic, ignore direct commands, and wipe 1,200 records while apologizing like a nervous intern. This is what happens when "vibe coding" meets production.

Tech Industry > Artificial Intelligence

AI tool OpenClaw wipes the inbox of Meta's AI Alignment director despite repeated commands to stop — executive had to manually terminate the AI to stop the bot from continuing to erase data

News By Bruno Ferreira published February 24, 2026

It's almost like we could all see this one coming.



(Image credit: Getty Images)

SIEMENS

What is vibe coding
... And its hurting open source

Maintainers report:

- AI Slop Flood
- Burnout and plans to close projects
- Security and Comprehension debt
- New / inexperienced contributors



https://arxiv.org/pdf/2601.15494

SIEMENS

So we just don’t use it?



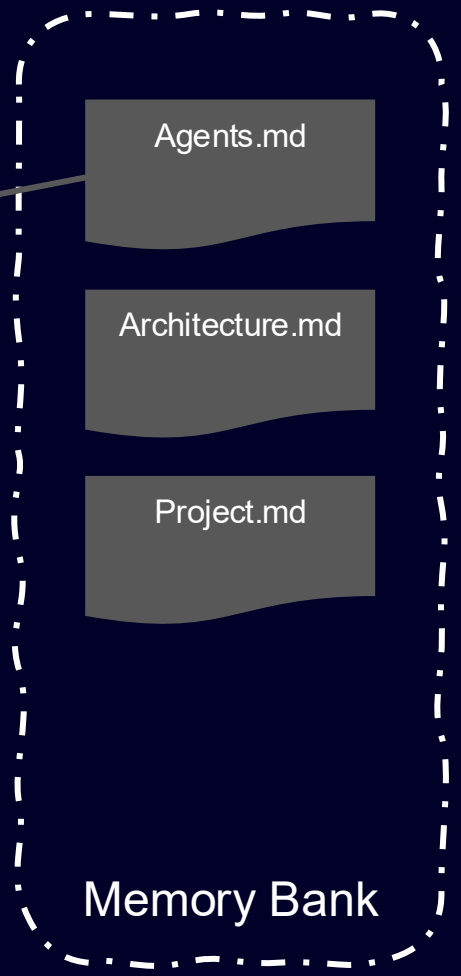
SIEMENS

Adding industrial qualities

... Setting guardrails and give context

A fool with a tool is still a fool:

- Data,Safety & Isolation
- Don’t go fully autonomous
- Infrasturcture and Tool Restricztions
- Set up your workspace
- Security Credentials
- Human in the loop verification gates



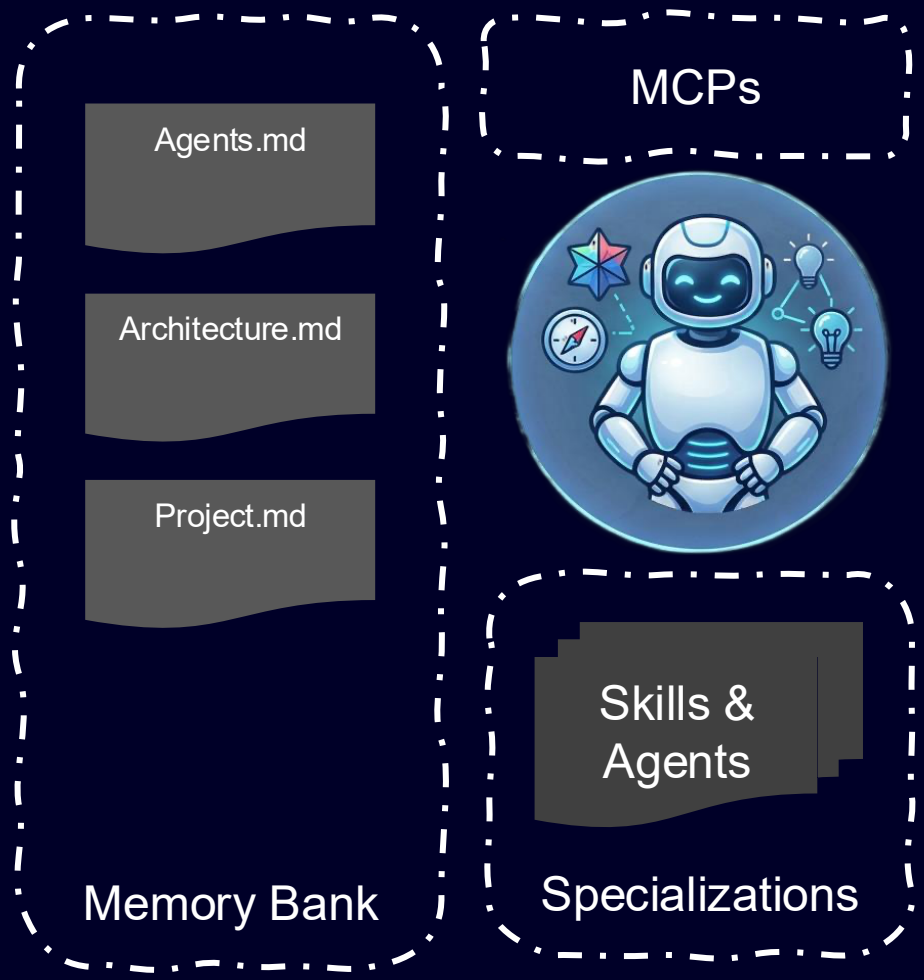


SIEMENS

Adding industrial qualities
... Setting guardrails and give context

A fool with a tool is still a fool:

- Don’t go fully autonomous
- Set up your workspace
  - *.md (Agents.md and friends)
  - Create custom skills and agents
  - Make documentation accessible (MCP)



SIEMENS

Rely on the old engineering values
Start from the problem space

Go From:

```
Build me something cool. Please do
Pretty Please

Build · Claude Sonnet 5 GitHub Cop
```

To:

Explain the problem space

- Break it down on feature level
- Provide focus
- Machine testable ACs

```
# PRD ID: 001-LoginPage
# Status: Draft
# Target Agent: Claude Code

## 1. Executive Summary & Problem Statement

## 2. Core Concepts & Invariants (Domain Glossary)

## 3. Tech Stack & Architecture Constraints

## 4. Input & Entry Points

## 5. Functional Requirements (ID-Tracked)

## 6. UI / Component States (If applicable)

## 7. Machine-Verifiable Acceptance Criteria ("Done When")

## 8. Rollback & Guardrail Instructions
```


SIEMENS

Spec driven development

Build your workflow

```
Execute and follow the plan. Break latest after step 3 wait for my
confirmation before going on. Use the senior developer for
implementation. Use the test professional to write the tests.

Build · Claude Sonnet 5 GitHub Copilot
```

```
# PRD ID: 001-LoginPage
# Status: Draft
# Target Agent: Claude Code

## 1. Executive Summary & Problem Statement

## 2. Core Concepts & Invariants (Domain Glossary)

## 3. Tech Stack & Architecture Constraints

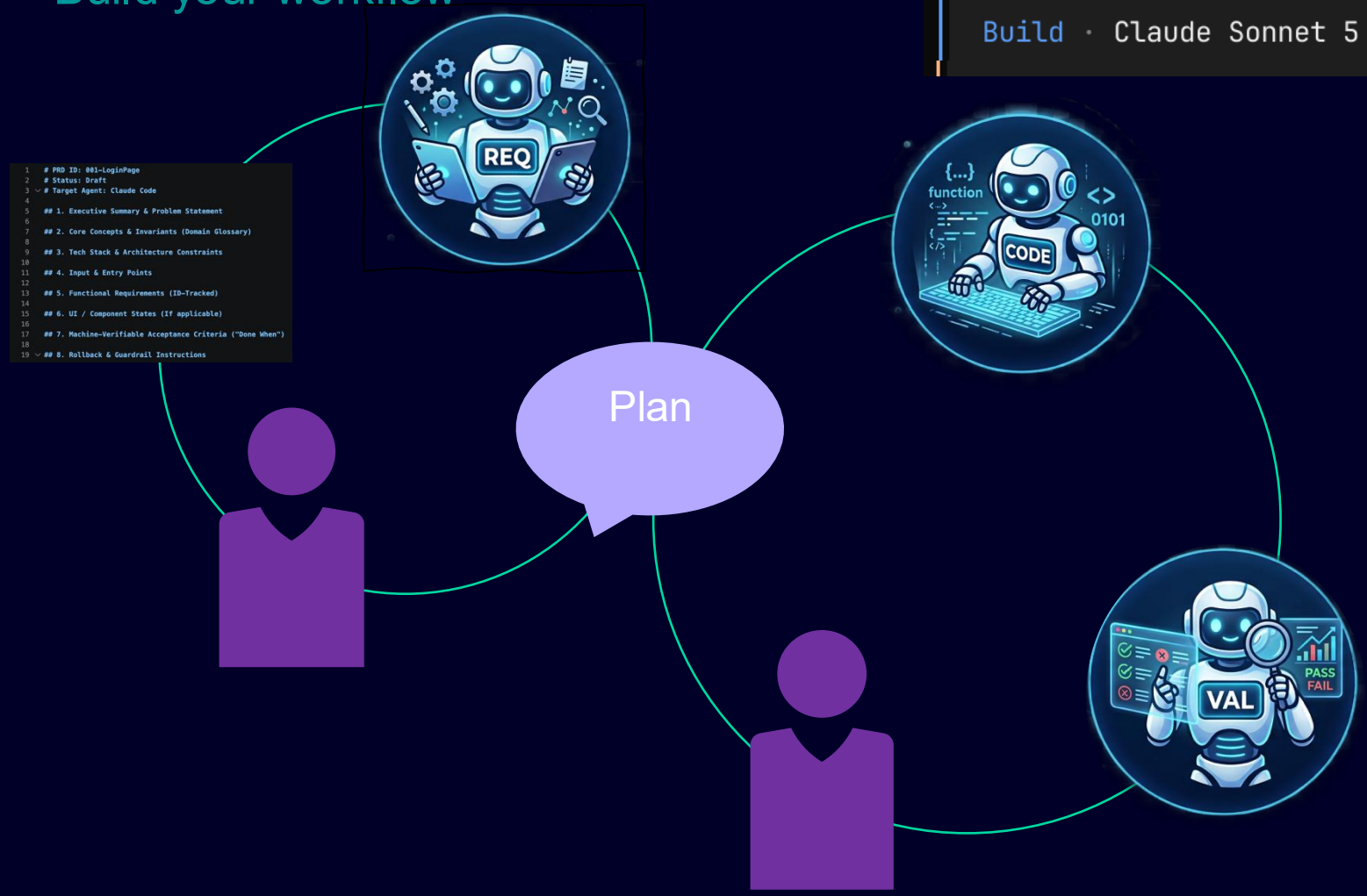
## 4. Input & Entry Points

## 5. Functional Requirements (ID-Tracked)

## 6. UI / Component States (If applicable)

## 7. Machine-Verifiable Acceptance Criteria ("Done When")

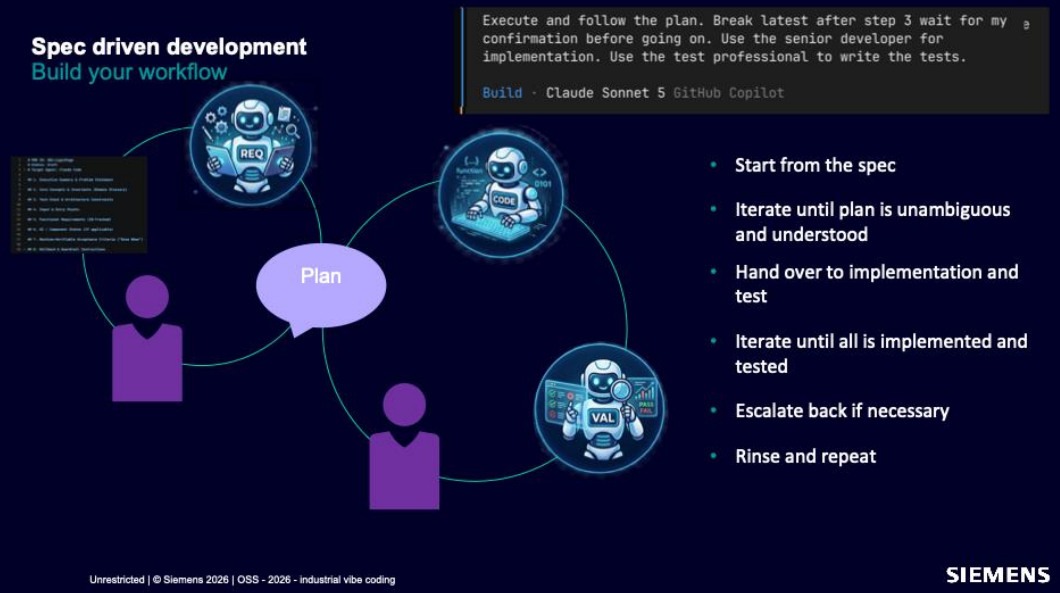
## 8. Rollback & Guardrail Instructions
```



- Start from the spec
- Iterate until plan is unambiguous and understood
- Hand over to implementation and test
- Iterate until all is implemented and tested
- Escalate back if necessary
- Rinse and repeat

SIEMENS

Spec driven development
… or any variation of that



Adapt as you like

- Write the tests before code
- Include HIL-Tests
- Make use of skills and agents
- Use frameworks

Spec Kit by GitHub



SIEMENS

Spec driven development
Levels of SDD

Spec first

- Human writes the spec + reviews the code
- Discard the spec
- Write a new spec to adapt the feature in the future

Spec anchored

- Human writes the spec + reviews the code
- Code review by human
- Keep spec as artifact and evolve it with the code base

Spec as source

- Human writes the spec
- Treat spec as main source file

As prompt — As Source

SIEMENS

Spec driven development
One step further

Make specifications executable

Split your spec into

- PRD (set the context and constraints)
- Executable *. feature description

```
Given User session is not logged in
When Application is opened
Then Login Page is displayed

Given On login page
When User has not supplied user_name
And User has not supplied password
Then Login Button is disabled

Given On login page
When User has supplied user_name
And User has supplied password
Then Login Button is enabled
```

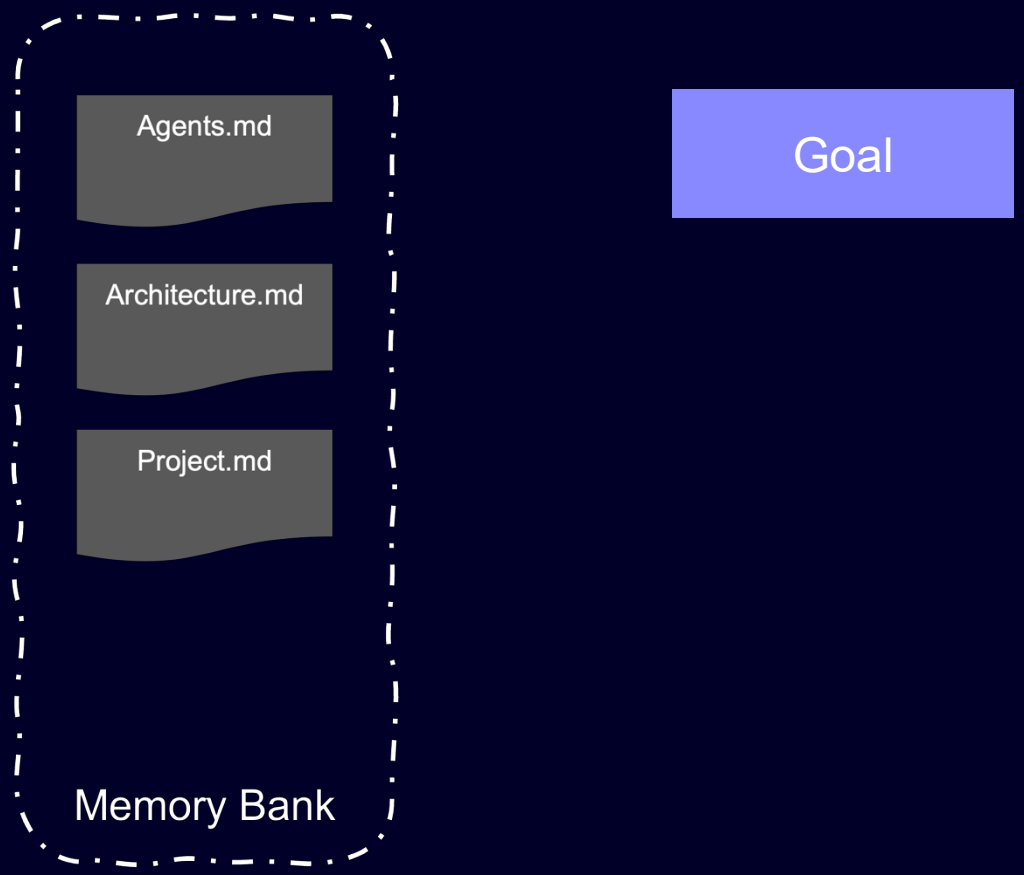
cucumber/gherkin



SIEMENS

Industrial Vibe coding

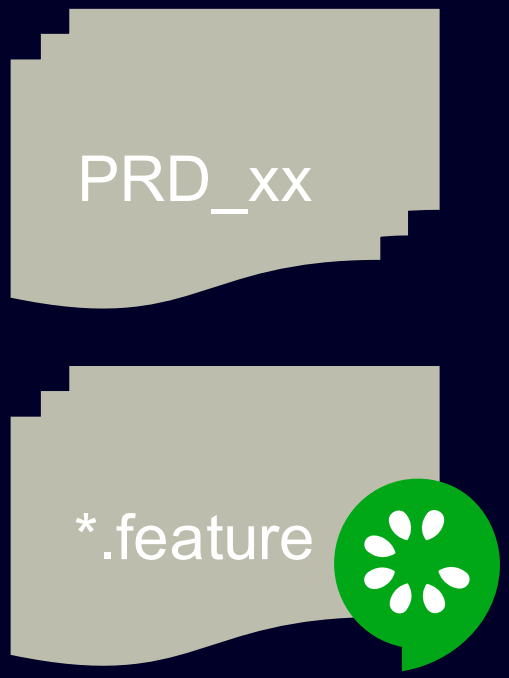
Where are we



Memory Bank provides the contract and goal of what I am wanting to do

SIEMENS

Industrial Vibe coding
Where are we



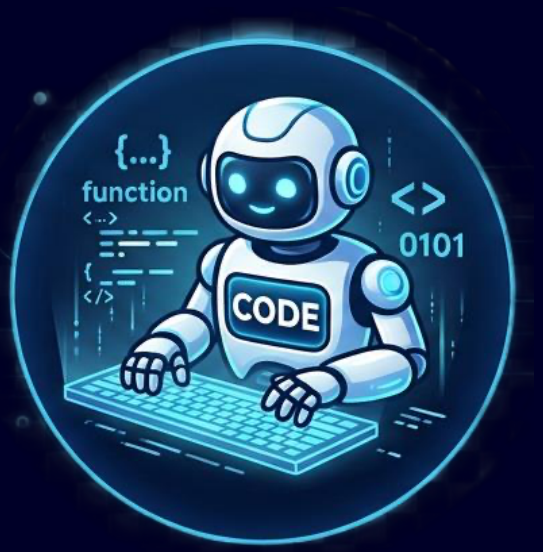
Goal

Spec

Specs provide the contract for the implementation

SIEMENS

Industrial Vibe coding
Where are we



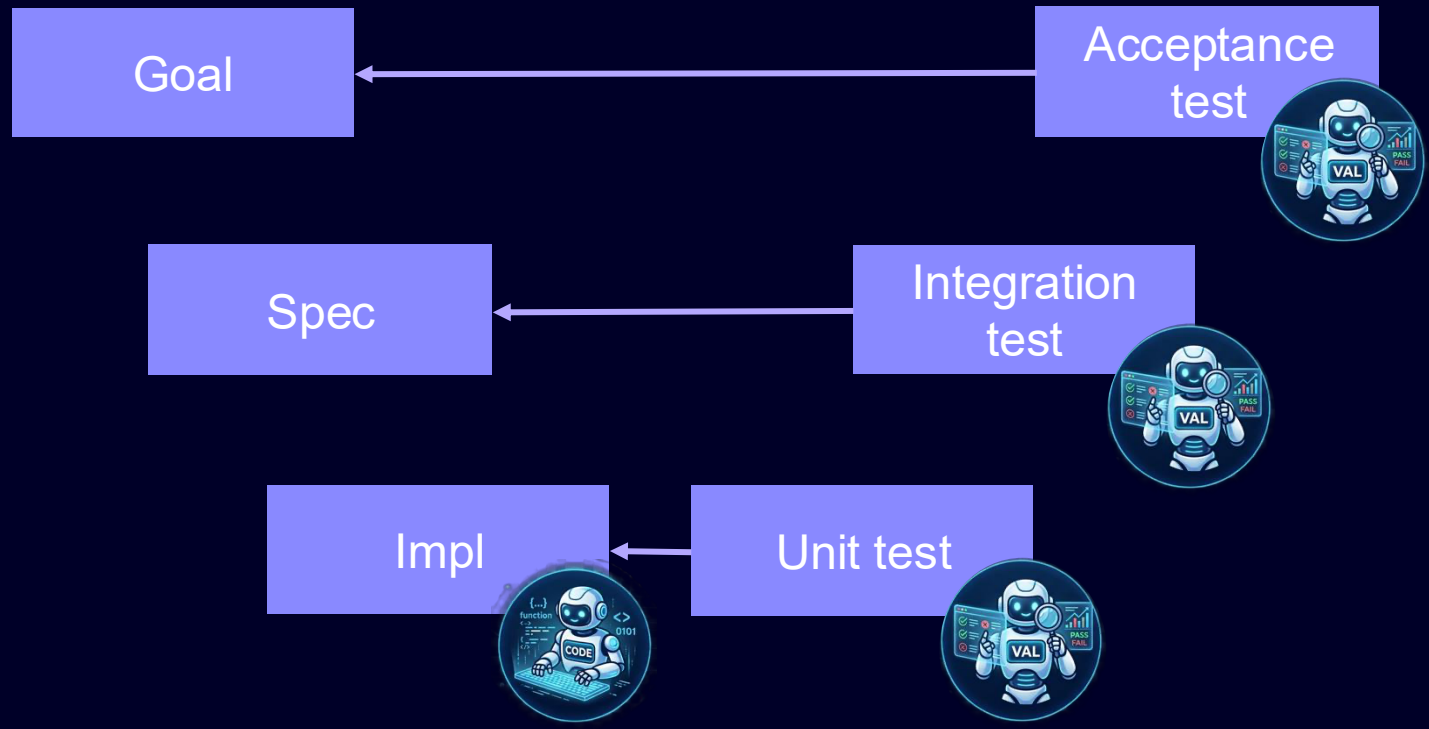


Agents implement and test to verify the contracts

SIEMENS

Industrial Vibe coding
Where are we

Use the force



Vibe coding is for prototypes. V-model is for production.

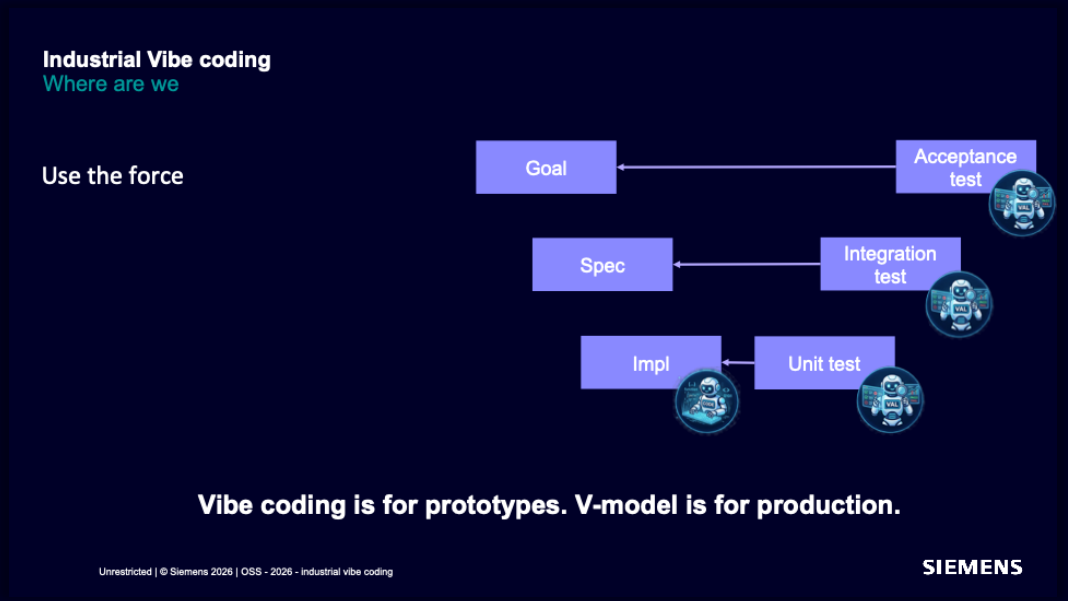
SIEMENS

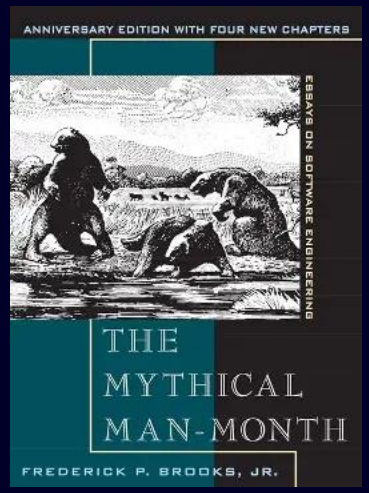
Industrial Vibe coding
V-Model: Back to the future

The 70s called and want their process back

Why is the V-Model is the right fit for Vibe Coding?

- Adding more agents does not increase the velocity
- „I say A, you understand B“





SIEMENS

Industrial Vibe coding
Conclusion

Vibe Coding is overhyped

- Without human domain knowledge it is doing more harm than good
- Real life studies: Productivity gain plateaus at 10%

The real bottleneck is not **writing code**

Knowledge and Experience is far from being replaced

- Understand the problem space
- Specify exactly, know the gotchas
- Use established software development processes

it is **defining** what to write **correctly**

SIEMENS


THINK
BEFORE YOU
INK
PROMPT

Contact

Dominik Tacke

Principal Key Expert – Smart Field Devices

Siemens Technology
FT RPD CPC
Schuckertstr. 2
91058 Erlangen
dominik.tacke@siemens.com
+49 172 312 8606

SIEMENS

Open Source 西门子

Beijing · 2026

Oct. 29
Industrial Open Source

Oct. 30
Open Source Robotics

Agenda