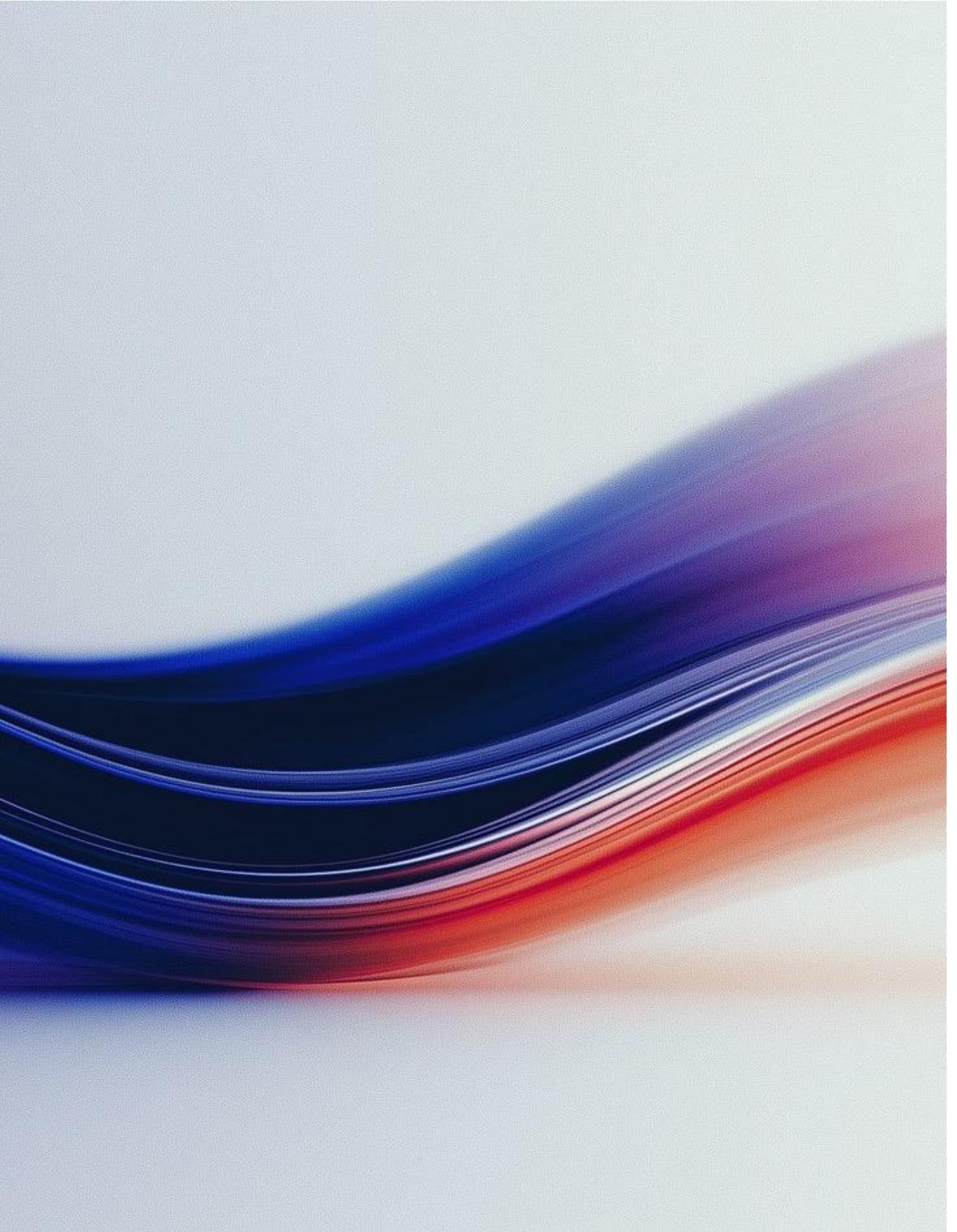
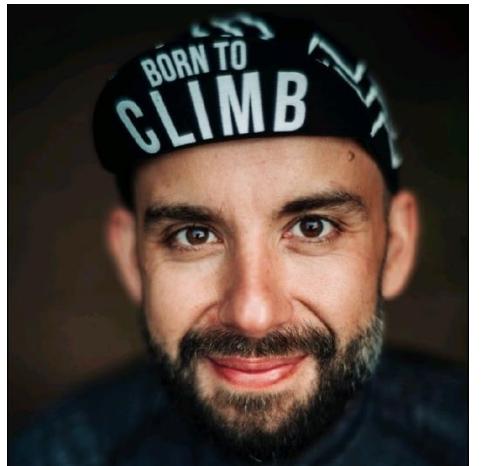


The Agentic AI Paradigm Shift in (Data) Engineering

The Agentic Shift: Scale Your GenAI with Success

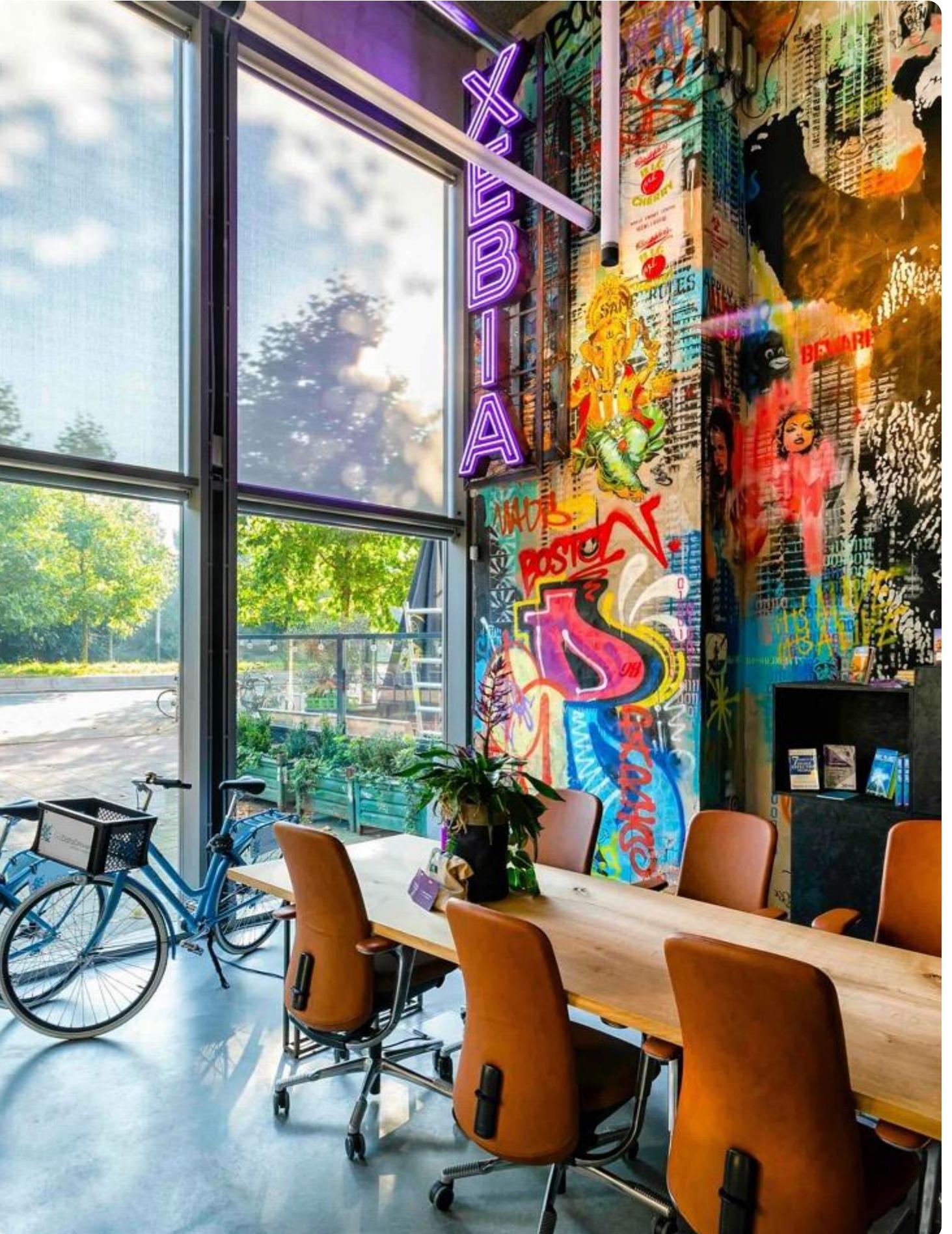




Marek Wiewiórka



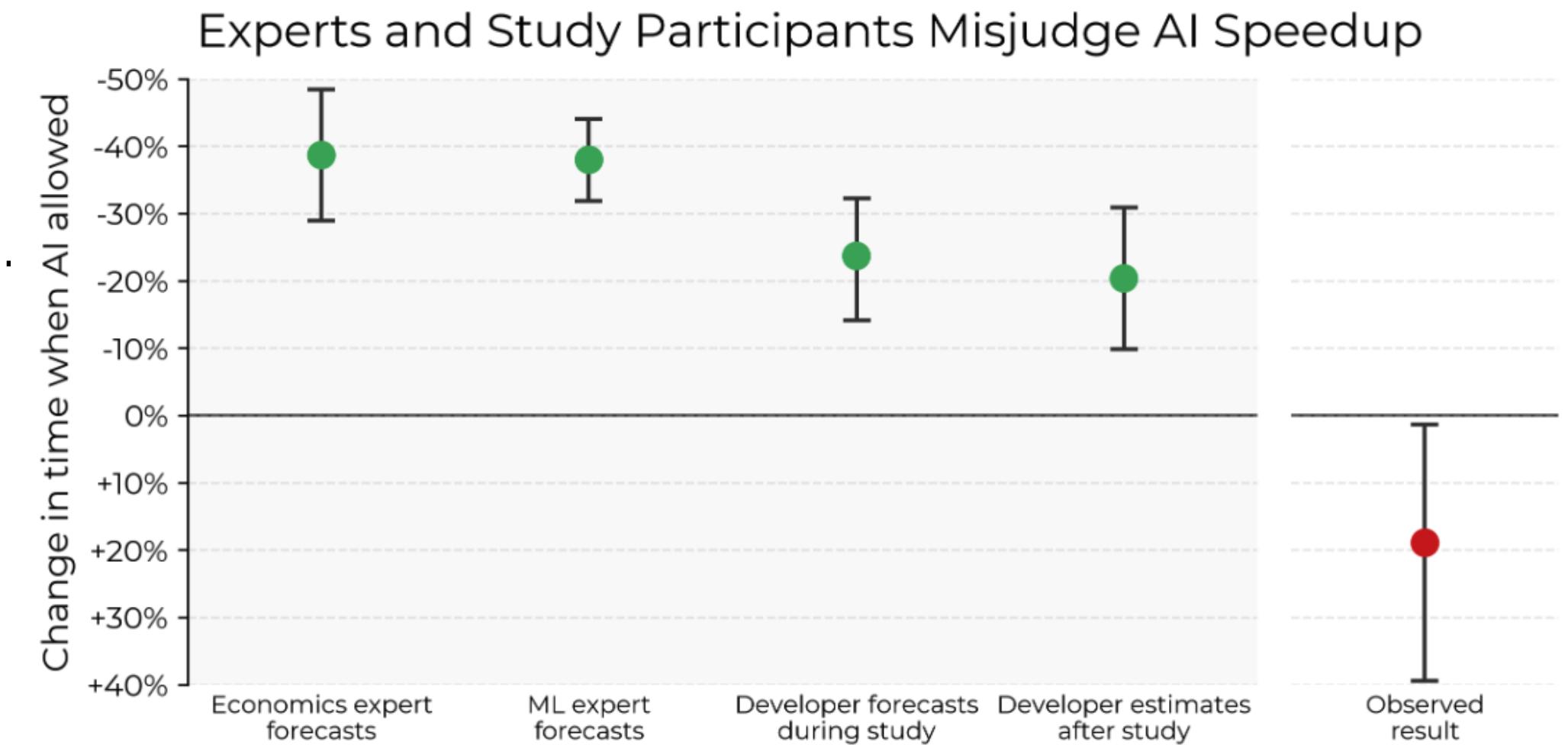
- 20 years of experience in data and analytics
- Chief Architect at **Xebia Poland**
- Leading **Xebia Data Research Lab**
- PhD in Computer Science, Bioinformatics
- Assistant Professor at **Warsaw University of Technology**, Institute of Computer Science
- A fan of road cycling and long-distance running



A good AI talk should start with a controversy and be followed by rising insight

LLMs **excel at** programming benchmarks but..

- **Over-estimated productivity boost**
- **Under-estimated human effort**
- **Lower-than-expected code quality**



Source: [Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity](#)

So, what's wrong here?

The Evolution of LLMs Capabilities 2024-2025

Memory (context windows)

From 1 page to 50 codebases at once

Reasoning (Chain-of-Thought)

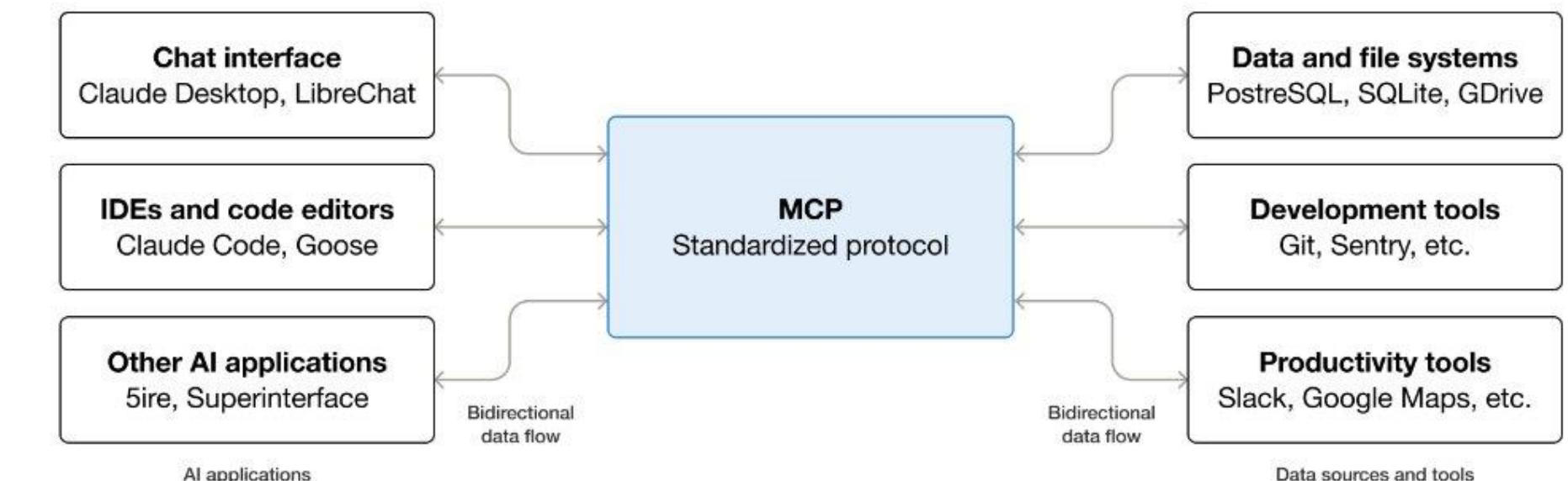
From "what usually comes next" to "let me think through this" (Sep 2024)

Function calling (external tools use)

From passive to edit, test, commit, deploy and tool interaction *standardization* and recent concepts like *skills* (early 2025)

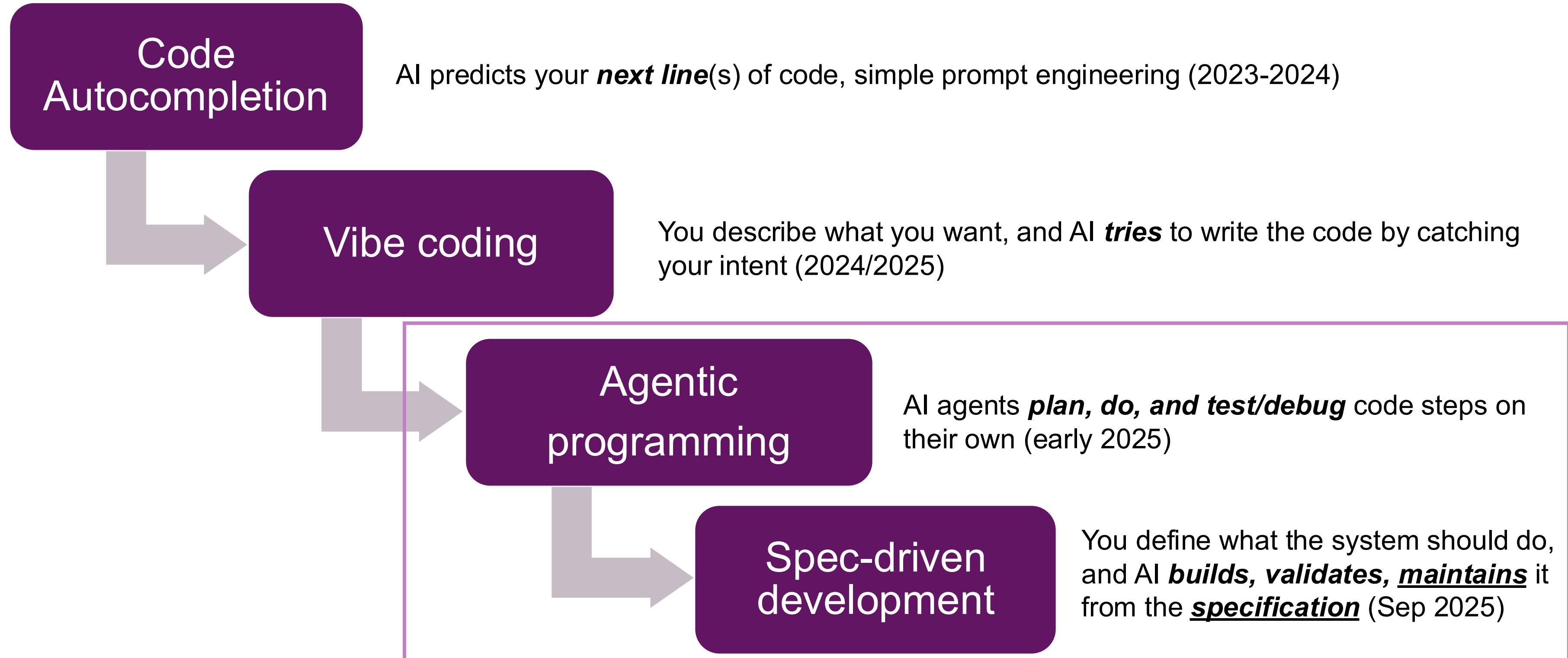
Tools Usage and Model Context Protocol (MCP), late 2024

- An **open-source standard** for connecting AI applications to external systems
- Lots of ready-to-use servers already available, including:
 - Data catalogs (e.g. DataHub, Gravitino, Atlan)
 - Databases, query engines, data platforms
 - Data tools (e.g. dbt, Airflow, Airbyte)

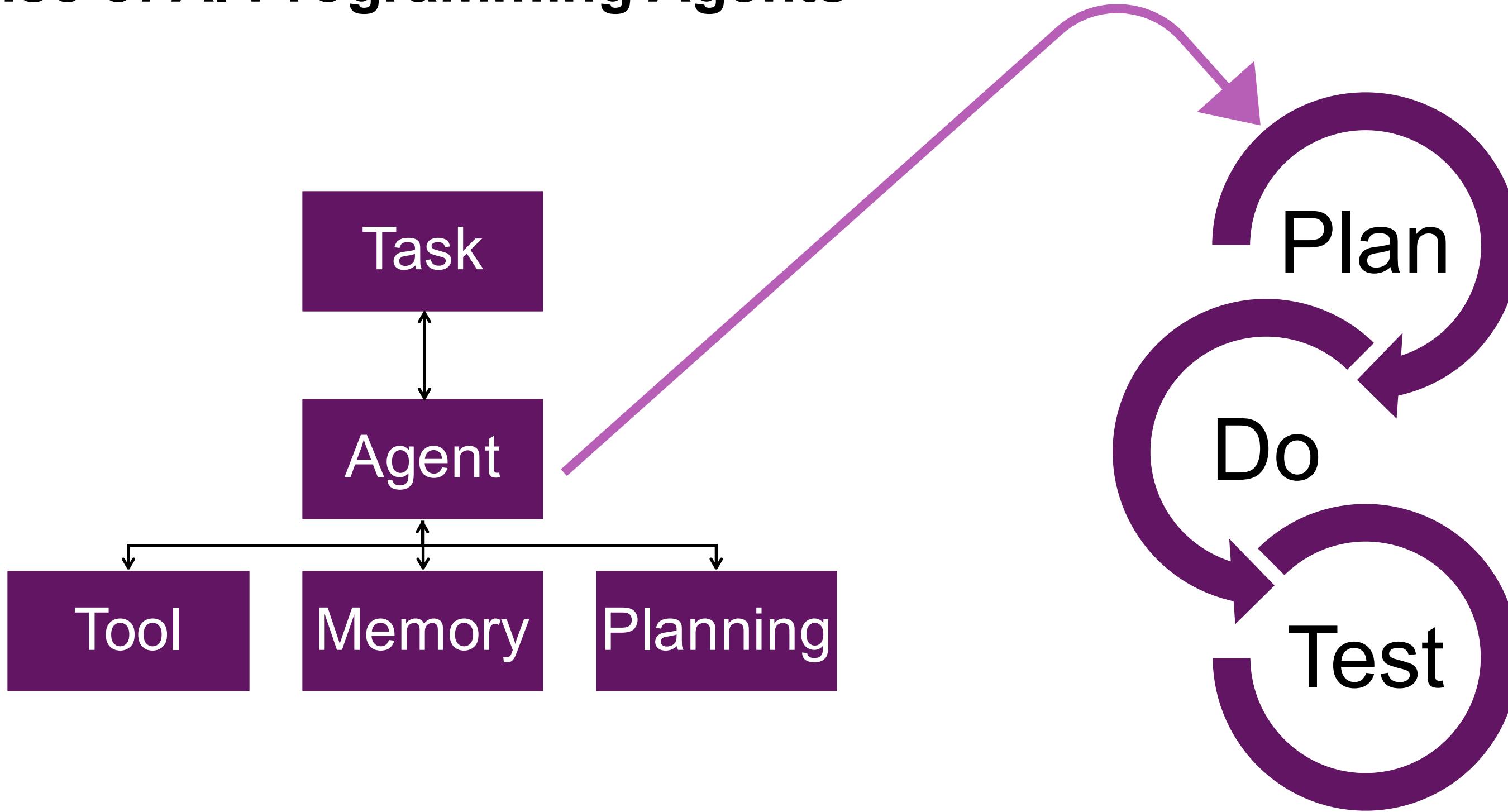


<https://github.com/modelcontextprotocol/servers>

The Evolution of AI-Assisted Programming

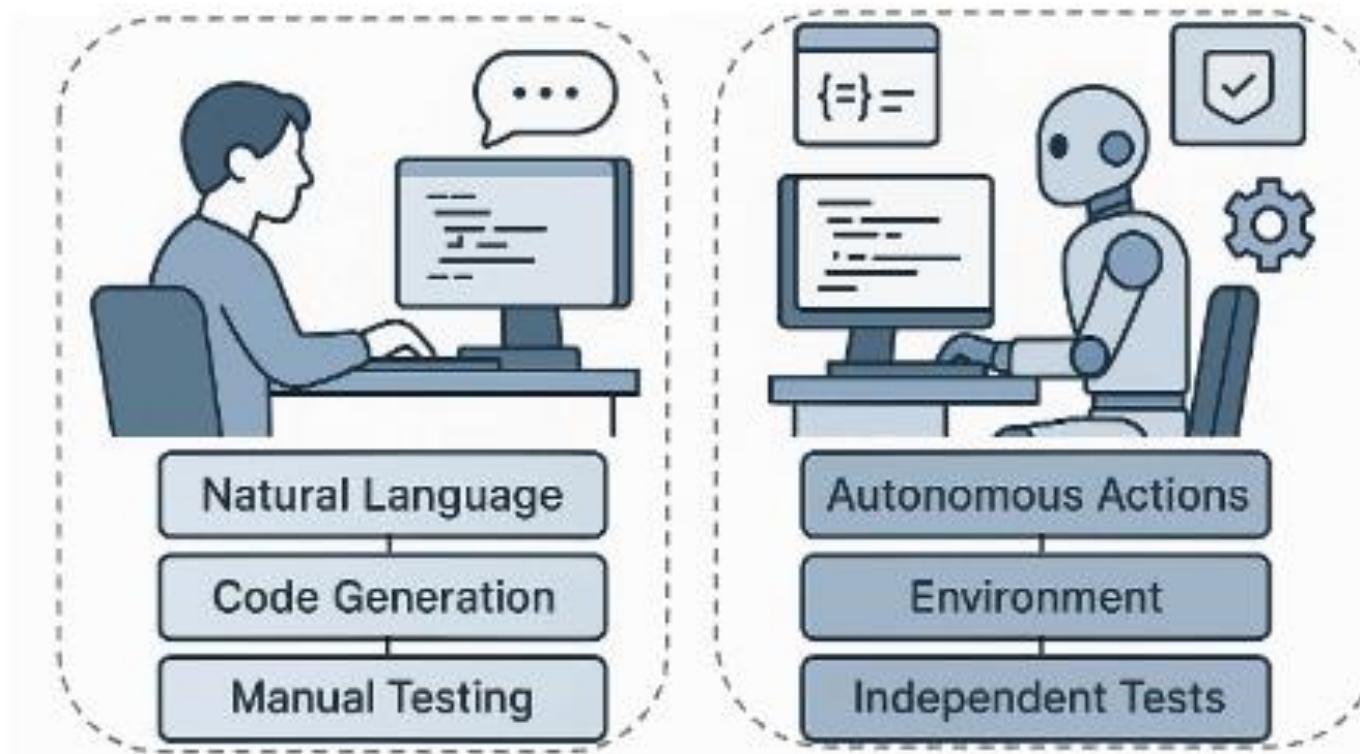


The rise of AI Programming Agents



Vibe Coding vs. Agentic Coding

- **Core Philosophy:** Human-AI co-creation vs. AI as autonomous executor
- **Planning & Task decomposition:** Human vs. AI
- **Tool Utilization by AI:** Code generation vs. Whole suite of development tools
- **Typical Task Scope:** Component-level vs. Feature-level
- **Developer's Primary Role:** Co-Pilot, Prompter, Tester vs. Architect, Designer



Claude Code CLI – February 2025

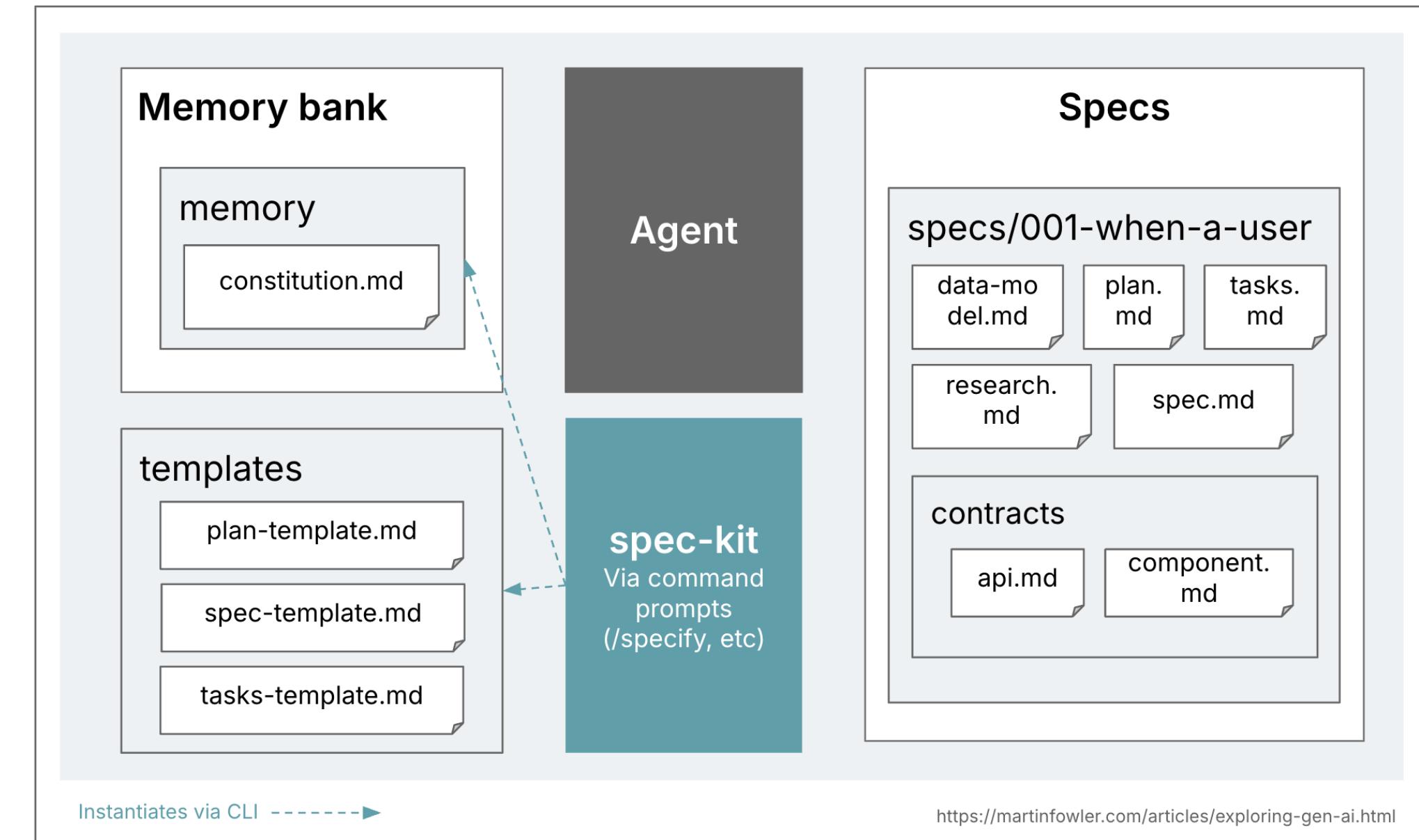
Codex CLI – April 2025

Gemini CLI – June 2025

Source: [Vibe Coding vs. Agentic Coding Fundamentals and Practical Implications of Agentic AI](#)

Spec-Driven Development (SDD) deep-dive

- ***Evolution*** not a Revolution
- Still not ***well-defined*** and ***overloaded***
- A set of structured artifacts written in NL for AI coding agents
 - memory
 - templates (mutable)
 - commands/scripts
- *Explicit* and gated (with checks) flow:



<https://martinfowler.com/articles/exploring-gen-ai.html>

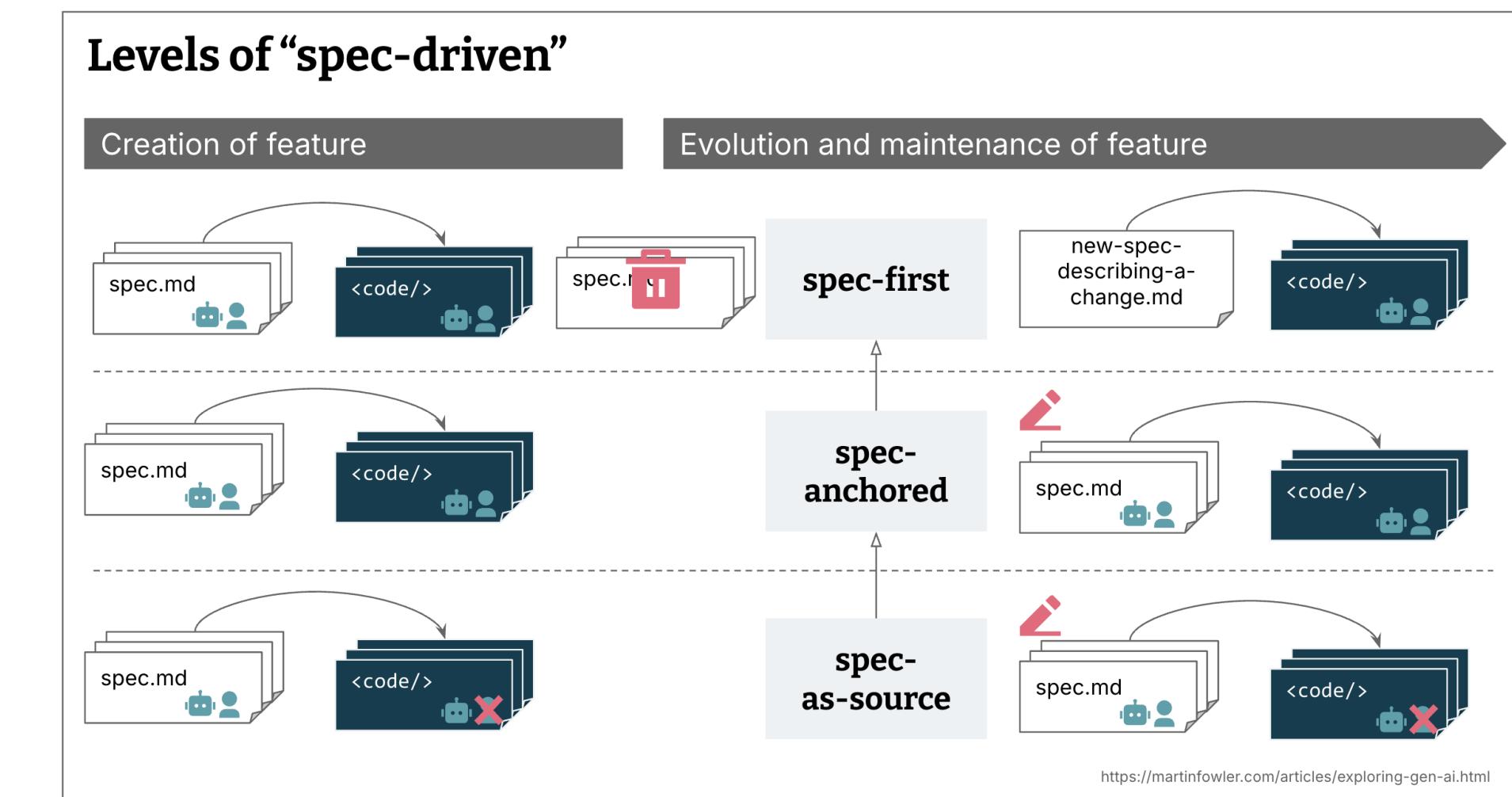
Specify (functional) → Plan (technical) → Tasks (decompose) → Implement (spec-kit)

Propose → Implement → Archive (merge) (openspec)

SDD - The Good, the Bad and the Ugly

- **Good:** (More) deterministic, structured - towards specification-as-source !
- **Bad:** Open standards not yet established, may introduce additional overhead
- **Ugly:** Lightweight adaptation for data domain (WiP)

Xebia Data Labs



AI-Assisted Development – The Gang of Three

- **Vibe coding** - rapid prototyping/PoCs, learning new tools/frameworks – lots of guesswork
- **AI-Agents Programming** – well-defined, *closed-form* problems, such as legacy code migrations, dependency upgrades, class refactoring, bug fixing
- **Spec-driven development** – new features and team collaboration, highly-reduced guesswork

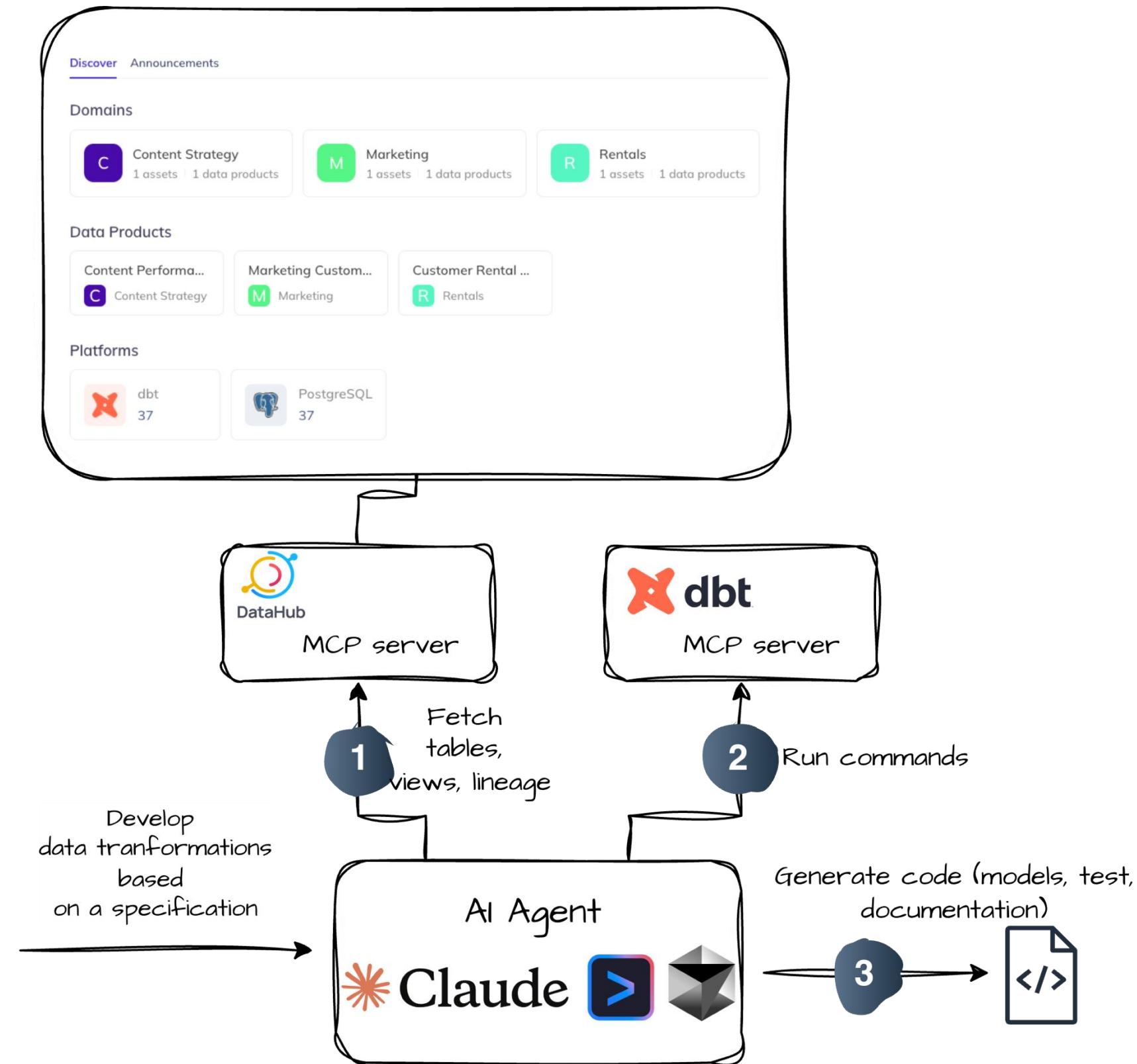
The AI Agents Paradigm Shift – opportunities and challenges

Enormous productivity when used properly

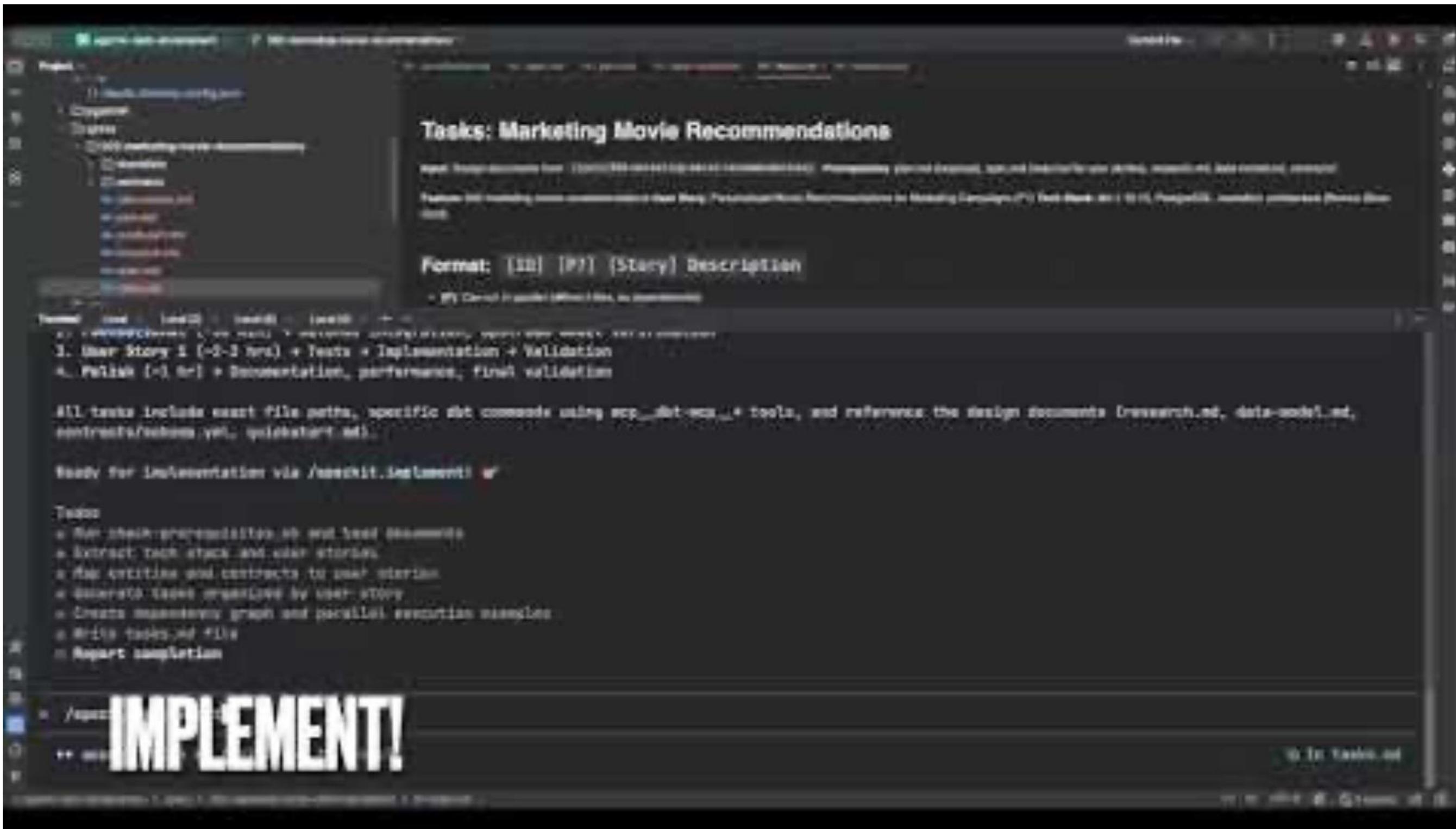
Challenges:

- **Core skills:** programming → architectural thinking, requirements analysis, specification writing, quality judgment
- **Goals:** technical artifacts (pipelines) → features (data products)
- **Roles:** engineer (technology) → product owner (business)
- **Way of work:** single-task → multi-task (frequent context switching)

AI coding/SDD Meets Data Engineering



Demo



Link: <https://youtu.be/jTDQ8jKZITY>

Conclusions and Looking Ahead

- Specification-as-source is just **around the corner**
- The **right tool** (AI-assisted coding “mode”) for the job
- A mindset shift is **as important** as advances in LLM technology
- Rethinking higher education and how we teach **engineers**

Thank you!

Got Questions?

October 28, 2025

Scalability of GenAI-based Systems - How to Achieve it?

