MARTINELLI
Keep IT simple

Spec-driven Development

Simon Martinelli

REPORT 15 SEP 2025

Kambundji shocks in Tokyo as Swiss peaks to take 100m hurdles gold

About Me

- 30 years in software engineering
- 25 years with Java
- Self-employed since 2009
- Teaching at two Universities
- Co-lead Berne, JUG Switzerland

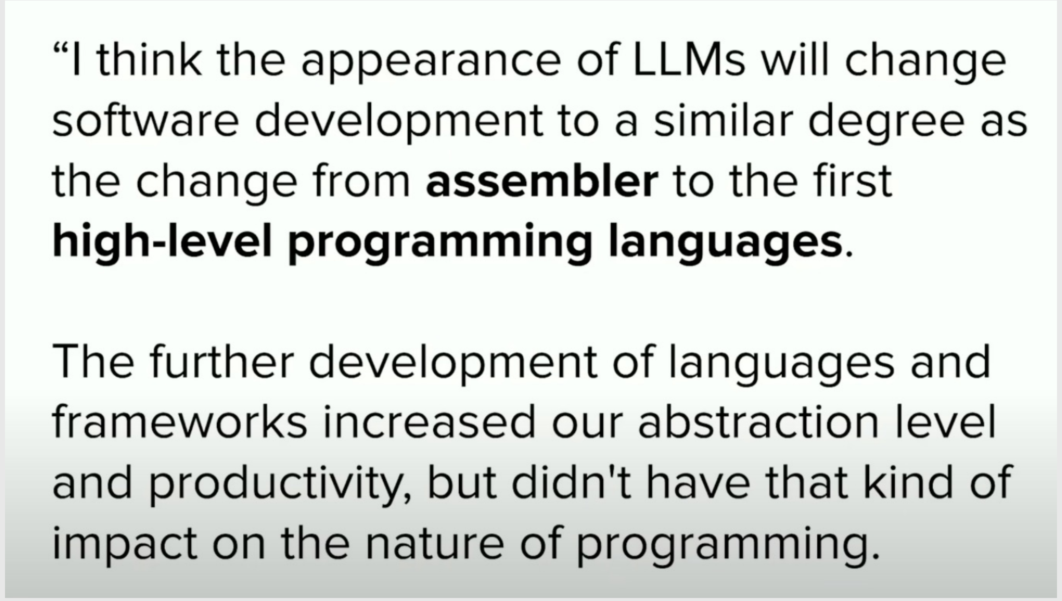
ACCIDENTAL PROMPT ENGINEER

Java™ Champions

vaadin}> Champion

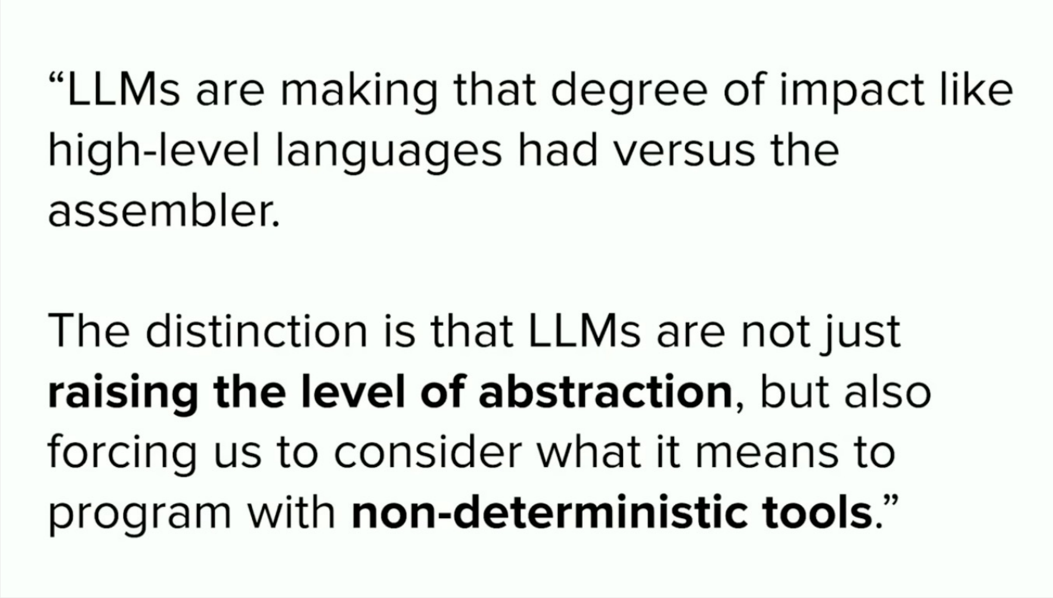
Oracle ACE Pro

Quotes by Martin Fowler



“I think the appearance of LLMs will change software development to a similar degree as the change from **assembler** to the first **high-level programming languages**.

The further development of languages and frameworks increased our abstraction level and productivity, but didn't have that kind of impact on the nature of programming.



“LLMs are making that degree of impact like high-level languages had versus the assembler.

The distinction is that LLMs are not just **raising the level of abstraction**, but also forcing us to consider what it means to program with **non-deterministic tools**.”

What is Vibe Coding?

I just see things, say things, run things, and copy-paste things, and it mostly works.
- Andrej Karpathy, OpenAI Co-founder, former Tesla AI lead, February 2025

- A **programming approach** where you
  - Use natural language to describe what you want
  - Let AI (LLMs) write all the code
  - Accept code without full understanding
  - Focus on the goal, not the code itself

Is Vibe Coding Any Good?

Good

- Prototypes
- Hobby projects
- Quick experiments
- Non-programmers building simple tools

Risky

- Production code
- Security-critical apps
- Complex systems
- Business applications

Caution: AI Is Not a Compiler

- **AI** code generation **isn't a compiler** - it's an assistant!
- Many developers expect AI to work like a compiler:
  - Precise input → perfect output
  - But that's not how it works
- AI is **not deterministic** and makes mistakes
- **You are responsible for the output!**

AI Native Development

- **Spec-Centric Development**
  - Clear intent/specs guide AI to generate meaningful code
- **Context-Aware Development**
  - AI agents understand full codebase context
- **Agent Experience (AX)**
  - Autonomous AI tasks (bug fixes, PRs, tests) enhancing developer throughput

https://ainativedev.io/

Spec-driven Development

- **Start with a spec**, not code
- Spec is the contract and **source of truth** for tools and AI
- Leads to less guesswork, fewer surprises, **better code**

- **Process**
  - AI Unified Process: https://aiup.dev
- **Tools**
  - Amazon Kiro: https://kiro.dev
  - GitHub Spec Kit: https://github.com/github/spec-kit
  - Tessl: https://tessl.io

https://github.com/simasch/ai-track-and-field

Recommendations

- **Scaffold** the application **yourself**
- Constantly **review** everything that is generated
- Create **test case examples**
- Use **MCP Servers**

Conclusion

- **Specs help** to reduce non-determinism
- AI feels like a teammate but **it's just a tool**
- It can accelerate development, but **you must**:
  - **Review**, **understand**, and **test** the output
  - **Know** your **architecture** and **domain**

Thank you!

- **Web**
  martinelli.ch
- **E-Mail**
  simon@martinelli.ch
- **Bluesky**
  @martinelli.ch
- **X/Twitter**
  @simas_ch
- **LinkedIn**
  https://linkedin.com/in/simonmartinelli