

Title: The AI Shift: Rethinking the Developer's Role in a Machine-Driven World

Speaker: Davide Montesin · 07/11/2025

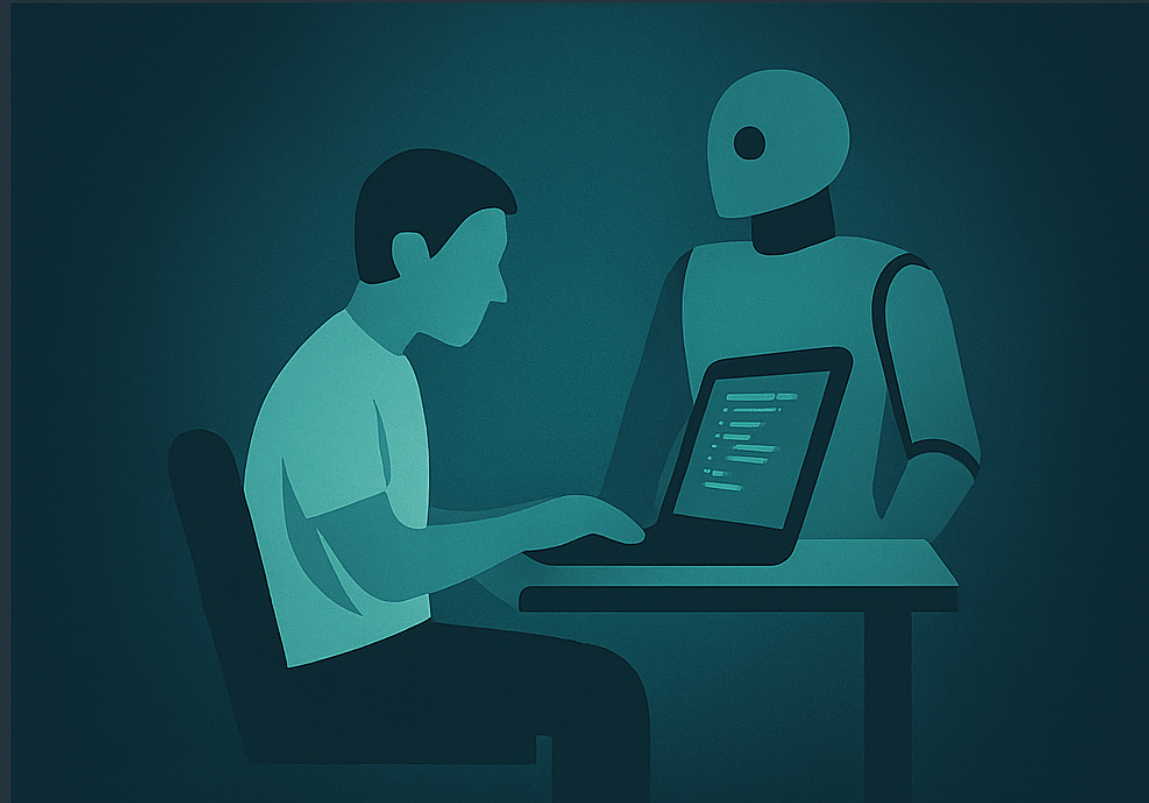


Title: A Lifelong Passion for Computer Science and Hacking

- I've gone from counting the bits and bytes that fit into CPU register in the '90 to working with AI.
- Always curious about how things work under the hood.
- Founder of Catch Solve, a tech company focused on software and data quality

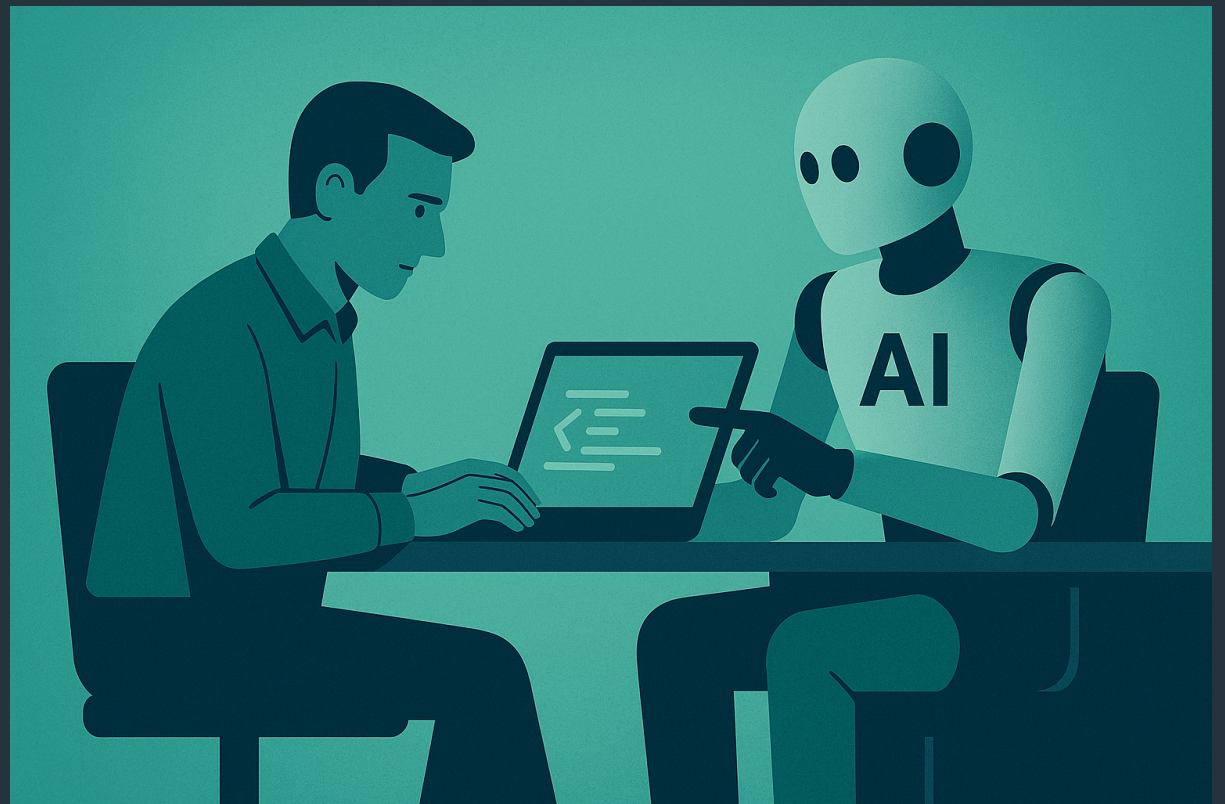
Title: The Current State of AI for Developers

Today most devs use AI tools — from GitHub Copilot to ChatGPT — to write, test, or understand code. A significant share of code in a typical project is already generated or assisted by language



But that doesn't mean a developer's job has become trivial. AI accelerates, but it doesn't understand context. It suggests solutions, but it doesn't know the product vision. And when it's wrong, it's often wrong convincingly.

This is why the developer role



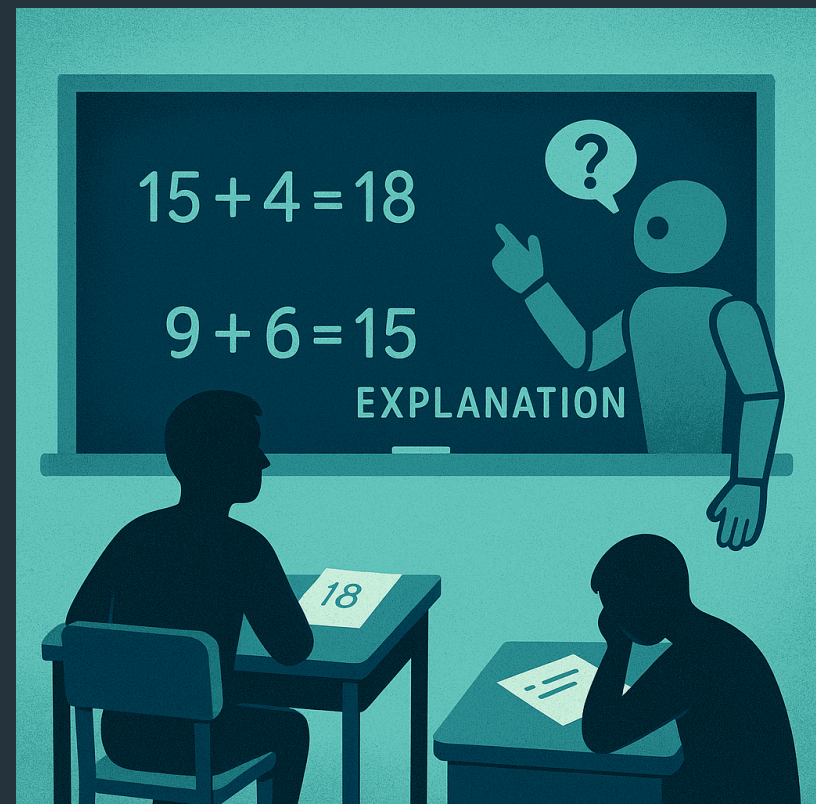
Title: Risks and Opportunities for Junior Developers

- Routine tasks now automated
- Harder to learn fundamentals by repetition
- Risk: losing the hard-won learning that comes from mistakes and from time spent



Title: How Juniors Can Benefit

For juniors, AI can be a powerful accelerator for learning. Like at school when you copy from the classmate next to you: you get the answer, not the understanding. Use AI to learn the why —



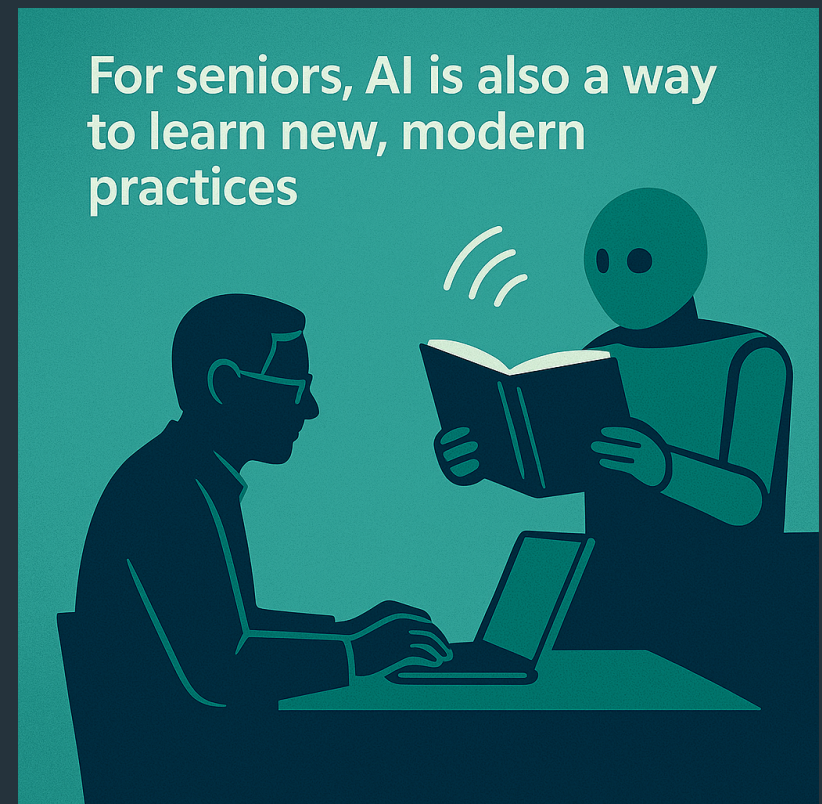
Title: Risks and Opportunities for Senior Developers

- AI can perform simple coding tasks, reducing operational workload.
- AI can explain, refactor, and even suggest better solutions.
- AI can generate documentation



Title: New Ways Ahead

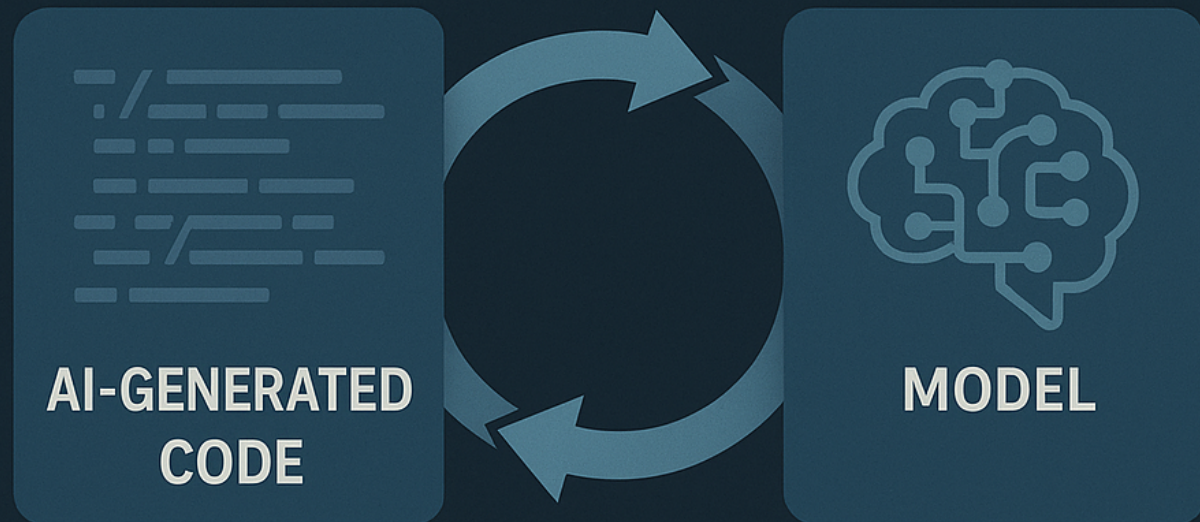
- AI corrects syntax; seniors bring experience and architectural vision.
- AI can't assess trade-offs, ethics, or long-term design.
- Seniors mentor on thinking, not typing.



Title: The Feedback Loop Risk

- Models trained on AI-generated code risk data degradation
- Bias amplification & quality decay

“If AI learns from human code today, and from AI’s code tomorrow... who will safeguard the quality of knowledge?”



Title: When Companies Run Too Fast with AI

- Many teams use AI-generated code they don't fully understand
- Short-term speed, long-term fragility
- When things break, companies need experienced developers/consultants



Title: When Security Is an Afterthought

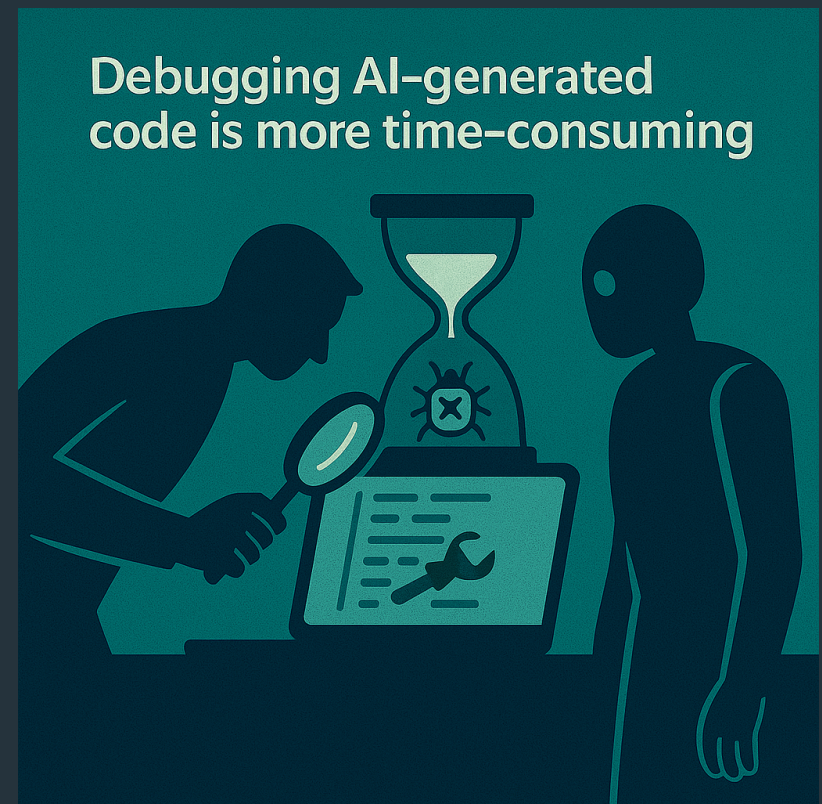
- As a consultant, I've also seen software with login security issues.
- Examples: weak password handling, missing rate limiting, poor session management.



- Examples: weak password handling
- Missing rate limiting, poor session management

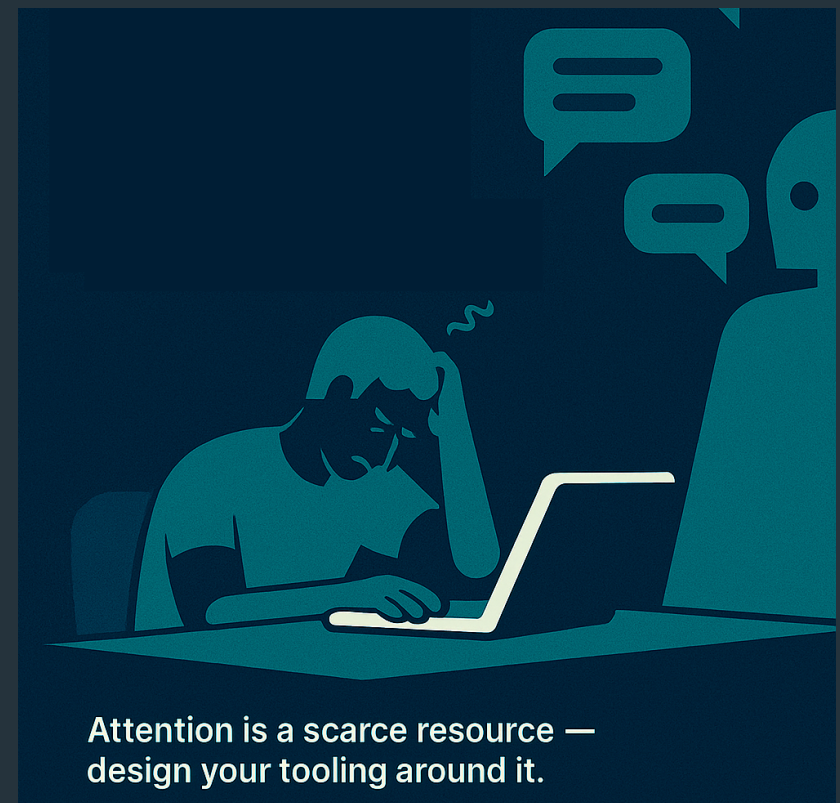
AI Tool Frustrations

- “The biggest single frustration, is dealing with AI solutions that are almost right, but not quite.”



Title: Helpful... until they aren't

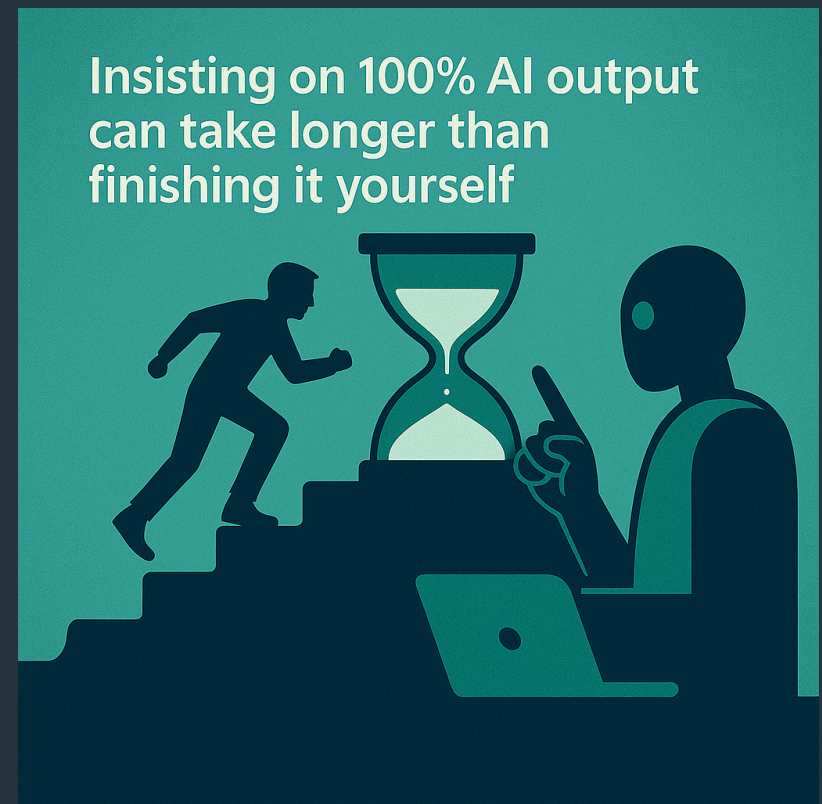
- Constant suggestions can interrupt deep work and flow.
- Cognitive overload from choosing among “almost right” options.
- Tip: use focus modes or temporarily pause



Attention is a scarce resource —
design your tooling around it.

Title: When forcing AI to do it all costs more

- Insisting on 100% AI output can take longer than finishing it yourself.
- Know when to step in: use AI for scaffolding, you for judgment and polish.
- Set a time/quality threshold: if it's not



The work is no longer writing lines: it's clarifying intents, constraints, and value. Code becomes a means, not an end.

- Conversations that generate systems.
- Specification before implementation.
- Time spent on the “why”, not the “how”.



The developer of the future orchestrates components, data, and models like a composer: a balance between intuition and rigor.

- More composition, less friction.
- Tools as an extension of thought.



Title: Practical principles for a human+AI workflow

- Keep humans as ground truth: tests, reviews, and data provenance first.
- Protect focus: pause suggestions when thinking; batch interactions.
- Security is non-negotiable: login, sessions, rate limits, secrets.
- Know when to stop: if AI stalls, take the wheel

Thank you ❤️