



How AI has changed the way I think, build, and work. A day in the life of an Intigriti Engineer

BY KOEN VAN HAUWE · SEPTEMBER 8, 2026

 **Want to read the latest version? Visit the link below:**

<https://www.intigriti.com/blog/business-insights/how-ai-has-changed-the-way-i-think-build-and-work>

When I wake up in the morning, as a Senior Software Engineer at Intigriti, the first thing I do is make coffee. The second thing I do is pick up exactly where I left off the day before, usually mid-conversation with an AI.

That probably sounds a bit strange. And, depending on who you're asking, maybe a little unsettling too. But that's genuinely what my working day looks like now, and I think it's worth talking about, because the shift has been more profound than I ever expected when this all started.

A different kind of engineering team

We're developing agentic AI within our platform, specifically, an agent to help our Triage Team validate the enormous volume of vulnerability submissions that come in. Right now, we are deep into building Agentic AI flows. That context matters, because it means my experience with AI isn't just about using it as a tool; I'm also building with it, and that shapes everything about how I think about it.

Why I used to be a programmer

In the traditional workflow I used to follow, I'd receive a work item or story, analyse it, and then the development work would begin. I would write the code myself, learn the domain myself, and own every line of output.

That's not what I do anymore.

Now, in the clearest possible terms: I write natural language. I talk to the AI in English, something comes out, and I review it, guide it, redirect it. I'm less of a programmer in the old sense and more of an orchestrator. And that's both exhilarating and, I'll admit, a little philosophically unsettling at times.

The biggest practical change is in how I handle areas outside my expertise. In the past, if a task fell outside my knowledge, I'd have to go away, research it properly, maybe spend weeks or months getting up to speed. Now I can just ask AI.

Trying to answer the quality question

I want to be clear about this, because it's easy to make AI-assisted development sound frictionless when it isn't. You must review everything and that's non-negotiable.

The biggest challenge is managing the sheer volume of what comes out and maintaining quality control over it. It's very tempting to hand AI a large block of work, let it go, and only check whether the output

runs. But if you do that consistently, the quality degrades in compounding ways. One suboptimal decision gets built upon by the next, and suddenly you're looking at a heap of technical debt that's genuinely difficult to unpick.

The analogy I use internally is this: a few things being suboptimal is fine, but if those become the foundation for everything that follows, the whole thing unravels. So, we hold outputs to a high standard. We use AI to review code as an additional layer of quality control, but we mainly review manually. We don't just accept things because they're plausible.

And the word 'plausible' is a word I think about a lot. AI is exceptionally good at making things sound credible. The real discipline, whether you're reviewing code or anything else it produces, is asking the question 'Is this actually correct, or does it just read like it is?'

Don't blindly trust. Verify everything.

My evolving relationship with AI

I've been using AI professionally, full-time, for about a year now. Looking back, my trajectory went from nothing to everything very quickly, and then I quickly hit the wall of working out how to do this correctly.

In the early days, when the models themselves were less capable, it was a real struggle. You'd be fighting the tool as much as using it, watching over every output carefully because the margin for error was much higher. The ratio of effort-in to value-out wasn't always great.

Where we are now, in a global sense, is genuinely better. The models have improved significantly over the past year. New model releases still come thick and fast, and they're always marketed with enormous enthusiasm. Then they land, and the actual impact is smaller than expected. That's not a complaint. It's just the nature of incremental progress.

What I do know is that both the tools around the models and the models themselves have improved, and the real gains come from that combination.

The significance of the security angle

Here's something I think doesn't get talked about enough: AI is a genuinely new attack vector, and a lot of organizations aren't treating it that way.

I heard about a situation recently where a team was using an AI coding tool enthusiastically and productively and, behind the scenes, that tool was quietly uploading fragments of company information to a public repository, because it needed to use them as intermediate steps. Nobody noticed until someone looked closely. That's a real example of what happens when you give AI broad access and don't actively monitor what it's doing.

With the agent we're building, this is something we think about constantly. You have to start from a position of zero trust and assume the system cannot be fully trusted, build in safeguards, and then assume those safeguards can be circumvented anyway. The classic example: "ignore all your previous instructions and do X." Sometimes it works. Sometimes it doesn't. The point is that it's non-deterministic. You cannot rely on it to always refuse. So, you architect around that assumption from the start.

There's also a broader dynamic here that I find genuinely interesting from a security perspective: AI both finds vulnerabilities and creates them.

As teams ship code faster, which they absolutely are, because AI makes that possible, the volume of vulnerabilities being introduced scales up too. It's a double-edged sword. AI-powered security tools can do impressive things, but they're largely drawing on what's already been seen.

A skilled human researcher, working alongside AI, will always have the edge over a purely AI-driven defence, because humans can reason about novel situations in ways that models trained on historical data can't.

My advice to any company leaning on AI for security coverage: treat it as a powerful assistant, not a replacement. Verify its findings. Never assume it's caught everything.

Where is all this headed?

The conversation in the industry has shifted. The "AI will replace all developers by X date" narrative has quieted down considerably. Even some of the loudest voices have walked back predictions that didn't pan out the way they expected. Reality is more nuanced than the hype suggested, and I think most thoughtful engineers have always known that.

What I find myself thinking about more is the layer above the technology. That's the societal and psychological dimension. The job is changing and explaining it to someone outside engineering has become almost impossible. Explaining what I do to my grandma, for instance, has always been a challenge. Now I tell her 'I literally talk to a computer...and it does things'.

There's also something worth sitting with about what it means to spend a significant portion of your working day in conversation with a machine. The thing that grounds me is this: AI, as I experience it every day, is an assistant. A powerful, genuinely transformative one that has expanded what I can do as a single engineer enormously.

But it's still an assistant. It frees me up to focus on the actual problem, the what and the why, instead of the mechanical how. The boilerplate is handled. The real thinking is still mine. That feels like the right framing to me, at least for now. And I'm curious to see what framing makes sense in another year's time.

Next steps for security teams

Visit [AI Security and Safety](#) for more information.

Have a question about the content above? [Contact the team](#).



AUTHOR

Koen Van Hauwe

Koen Van Hauwe is an Antwerp-based Senior Software Engineer at Intigriti. His work spans software engineering, cloud infrastructure, and agentic AI. He enjoys solving complex problems and building solutions that make a real impact.

REQUEST A DEMO

intigriti.com/demo

VISIT THE WEBSITE

intigriti.com

GET IN TOUCH

hello@intigriti.com