ORG DESIGN SUMMIT 2026 · AMSTERDAM

LLMs Made Building Easy — and Thinking Harder

Maxim Schepelin · Solution Architect, Booking.com

linkedin.com/in/schepelin

em-compass-book.com

GETDX · STATE OF AI IMPACT · Q2 2026

52.7%
code authored by AI

90%
adoption across tech

4–6 hrs
saved per engineer / week

getdx.com/report/State-of-AI-Impact-in-Engineering-Q2-Report/

Do these gains show up in revenue?

GETDX · Q2 2026 · SAME REPORT

×29

AI spending, in 12 months

–4.2%

ease of delivery

+50%

change failure rate

getdx.com/report/State-of-AI-Impact-in-Engineering-Q2-Report/

"…whether saved hours are being reallocated to product innovation or if they are simply being absorbed by existing organizational friction."

— GetDX, Q2 2026

Individual productivity goes up.

Organizational productivity stays flat.

BOOKING.COM

Two-year journey of adopting AI everywhere.

01 What AI didn't improve.

02 Where it's still taking time to get the benefits.

03 Where AI helped from day one.

Maxim Schepelin · Solution Architect, Booking.com

PART ONE

What didn't improve

MICROSOFT · 2009

1 in 3

experiments actually improved the metric they aimed to improve

Online Experimentation at Microsoft · researchgate.net

Iterate fast ≠ ship fast.

Iterate hard. Test ideas. Throw away the useless ones.

Experimentation is a gate.

A/B testing is a commodity. Make it part of your SDLC — and don't launch what doesn't move key metrics.

Guardrails: transactions · cancellations · customer-service requests.

Without guardrails, teams optimize locally-defined metrics at the cost of the ones that pay the bills.

Every feature you ship
is a feature you maintain.

Total Cost of Ownership = Development + Maintenance.

Maintenance is everything that keeps a system up and running.

THE MONTHLY BILL

20+ years in production · million lines of code

Infrastructure — compute, storage, licenses. It never stops.

Compliance — obligations that only grow, year after year.

Firefighting — on-call rotations, around the clock.

It takes a village — it’s full-time job to keep a 20-year-old system running.

The coordination tax — half the company depends on it, every change is a negotiation.

Building is a one-off payment. Maintenance is a subscription — for 20 years.

PART TWO · WHERE IT'S TAKING TIME

So we tried.

"Hey Claude, help modernize that legacy system…"

Which of the three frameworks should it use?

How to expose an API publicly — there are seven different ways here?

Which services are safe to depend on? Which are to be deprecated?

LLMs are extremely capable

when they have sufficient context.

In huge legacy systems, that context doesn't exist in written form. It's tribal.

The gap isn't tooling. The gap is governance:

standards · sources of truth · deprecation policy · architectural alignment

What's actually moving the needle

01 Decision trees — which database, when; how to expose an API. Guide that an engineer or an agent can follow.

02 Overlap mapping — find overlapping services, and decide what to phase out and what to evolve.

03 Guardrails — the business metrics, as hard constraints on every modernization.

Best practices didn't spread with the tools.

When Cursor arrived, the pilot had a waitlist. Same with Claude Code. Same with Codex. The tools spread in weeks.

But people kept reinventing solutions others had already figured out.

The fix: a central, discoverable skills repository.

A solution is figured out once — published, found, reused.

Still taking time.

Budget years, not sprints.

Recreate context from the past — write down what the organization knows.

Define the technical strategy for the future — which direction, what gets phased out, what gets built next.

None of it is model work. All of it is organization work.

PART THREE

Where AI helped from day one

CASE 1 · FINDING KNOWLEDGE

BEFORE

Every "single source of truth" project quietly failed — no one tool fits an account manager, a designer, and an engineer equally well.

AFTER

Ask a question, get an answer with sources. "What's an ADR?" takes seconds. Nobody cares where information lives anymore.

DIRECTION

Stop forcing one source of truth. Make all of it machine-findable.

https://www.glean.com/resources/customer-stories/booking-com

CASE 2 · ANSWERING DATA QUESTIONS

BEFORE

"What's the adoption of our loyalty program in region X?" — SQL skills, or a ticket into the analyst backlog: planning, prioritization, headcount debates.

AFTER

Ask an agent — it queries, visualizes, slices. Insight became a commodity.

WHY IT WORKED

Years of prior data governance — sources of truth, metric definitions, documented assumptions.

The agent is a component. The governance is the key.

AI amplifies
what you already have.

Capabilities and dysfunctions alike.

Foundations existed → a superpower.

Foundations missing → still paying the toll.

Mature capabilities + AI = Superpower.

modular architecture · reliable tests · observability · clean and governed data · experimentation gates

ORG DESIGN SUMMIT 2026 · AMSTERDAM

Shipping more won't get you there.

Solid foundations will.

Maxim Schepelin · linkedin.com/in/schepelin · em-compass-book.com