AI · DELIVERY · BROWNFIELD

Writing code got faster. Why didn't the work?



AI didn't remove the bottleneck, it moved it. In an existing system, the real cost is finding and proving the rules.

Muhammet Şafak • October 3, 2026

SECTION 01

Bottleneck

AI didn't remove the bottleneck, it moved it.

ONE STATION

It didn't vanish. It moved.

Make writing 3× faster, but leave review and verification where they are, and you haven't tripled throughput.

THE AUTHOR'S OBSERVATION · ONE TEAM I WATCHED

Output went up; throughput went down.

~2×

Code output

Roughly double once they leaned on agents

6

Weeks later

They were shipping more slowly

3×

PR queue

The pull-request queue tripled

2

Incidents

Traced back to changes nobody had read

DELIVERY PIPELINE

Throughput follows the slowest station



`got faster`

WHERE DID THE BOTTLENECK GO?

It was always expensive; now it's drowning

Review

Generation produces PRs faster than humans can give them real attention; review degrades into rubber-stamping.

Verification

Writing tests is nearly free; suites fill up with flaky, timing-dependent cases.

Integration

More code means more coupling and more ways for the same message to arrive twice and corrupt state.

DORA 2024 AND 2025 REPORTS, AS CITED IN THE POST

AI is an amplifier, not a shortcut.

Teams with strong delivery practices get more out of it; teams without them ship their dysfunction faster.

WHEN DOES THIS NOT APPLY?

Know which world you're in

Greenfield, prototype, spike

The constraint is still production

- ✕ No downside to protect yet
- ✕ Go fast
- ✕ No process for a system not yet built



Team, review queue, production

This is where the shift bites

- ✓ **There's something to keep honest**
- ✓ **Other people depend on production**
- ✓ **The bottleneck moves past the writing**

SECTION 02

Brownfield

So what about a system that's been running for years?

WHERE THE KNOWLEDGE LIVES

Greenfield vs. brownfield

CRITERION	GREENFIELD	BROWNFIELD
What is "correct"?	You define it	The system already defined it
Where is the knowledge?	With you and whoever made the request	In the code, the data, people's heads
Cost of a mistake	Reversible	Live data and users pay
Dominant cost	Production	Verification

What moves isn't code, it's rules.

In brownfield, what moves isn't code, it's rules. An agent is a multiplier where the knowledge is with you, and an assistant where the knowledge is in the system.

WHERE RULES HIDE

Three places, three ways to dig

Code says what a rule is; usually only people know why.

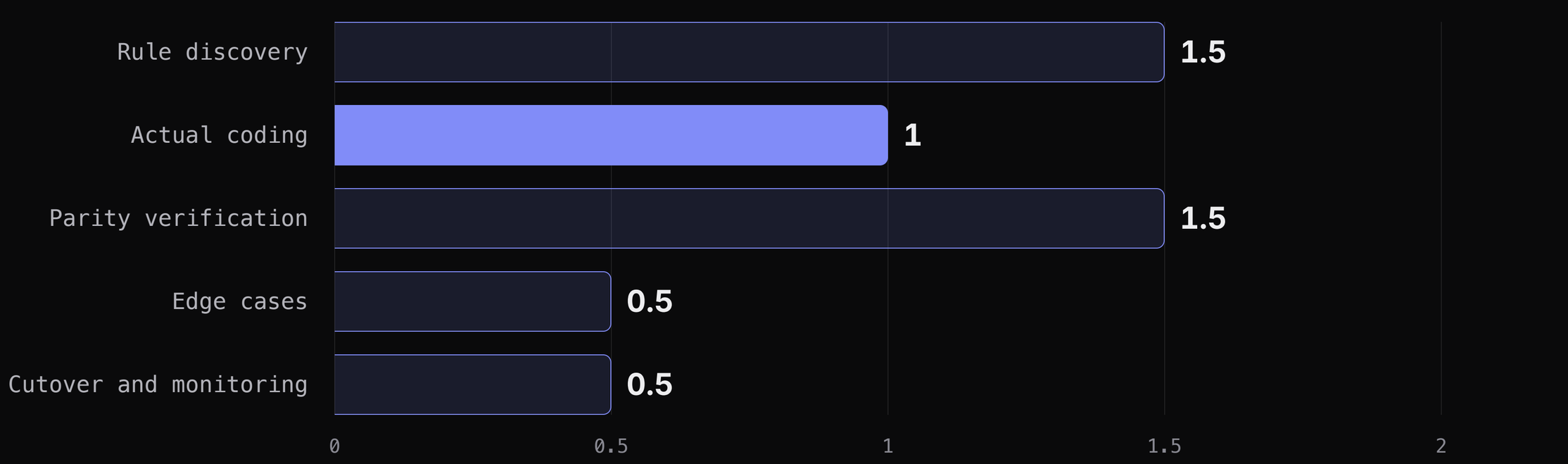
Unwritten exceptions in the code — 3 business rules in 15 lines

Implicit rules in the data — status counts, what NULL means

The reasoning in people's heads — was it meant to be this way?

AN HONEST BREAKDOWN OF ONE WEEK

Writing code is one day out of five



Approximate breakdown, a single example: migrating a CRM module.

WHERE AN AGENT IS A MULTIPLIER, AND WHERE IT'S AN ASSISTANT

What it speeds up and what it doesn't

Mechanical conversion

Turning one pattern into another; minutes, not hours.

Reading code

Extracting rules; the output is a list of questions, not an answer.

Verification tooling

Scripts that compare old and new, characterization tests.

Validity of a rule

An agent finds the rule; it can't tell you whether it's still wanted.

Verifying

Showing it produces the same results as the old one is the daily work.

Cutover and coordination

Not technically hard; but it takes up calendar.

Four rules

- [ ] Count unknown rules, not lines.
- [ ] Spend the first half day on discovery, then give the estimate.
- [ ] Budget at least as much for verification as for production; the ratio comes out around 1:1.5.
- [ ] Write the cutover as its own line item.



> Writing code got faster. Understanding what the years mean did not.

— **Muhammet Şafak** Fast from scratch, slow in a live system

Thank you

muhammetsafak.com

Sources: Engineering and Blog posts on muhammetsafak.com

