GenAI Coding and the Metrics Gap

Higher Productivity / Slower Delivery

What DORA Explored

DORA 2024 reveals a GenAI development paradox

75% of Developers report higher productivity

Delivery throughput down by 1-2% / stability by 7%

What gives???

Developer Metrics

Developer metrics focus on local optimization and easy measurement…

- More lines of code
- More bugs squashed
- More PRs pushed

Developers want to be on the right side of the metrics…

(We are looking at you, stack ranking)

Delivery Metrics

Delivery metrics are more objective

- Release cadence
- Patches per release

Feature sizes may vary, but the picture is consistent over time.

GenAI coding means:

- Higher developer velocity
- Lower delivery velocity

Types of Bug

Bugs made by humans:

Syntax bugs

Semantic bugs

(mitigated by tools / tests…)

Bugs made by GenAI:

Design bugs

Hallucinations

(mitigated by humans…)

AKA Technical Debt

GenAI Complacency Loop

The loop:
- Code looks correct
- Unit tests pass
- So it must be good, right?
- Was fine the last few times
- Metrics / management pressure

Faster code accumulation means faster Tech Debt accumulation…

…if developers stop paying attention

Which they do

GenAI — Love it or Hate it

39% of developers report little to no trust in AI

But development pressures mean they must use them

We can help with GenAI use:

- Best practices
- Strengths
- Weaknesses
- Failure modes