

AI-NATIVE SOFTWARE ENGINEERING

AI-NATIVE SOFTWARE ENGINEERING

A 3-Hour Interview Case Study
and Independent Reconstruction

CONTEXT → AGENTS → EVALUATION



AI shifts the bottleneck from coding to the delivery system

The fastest part of the workflow is no longer the only constraint.

WHAT CHANGES

- Implementation gets cheaper and faster.
- Review, integration and deployment absorb the new volume.
- Security and reliability become design concerns.



REBALANCE

HOW THE MODEL RESPONDS

- Define before agents execute.
- Use agents inside bounded workflows.
- Verify semantics, not just syntax.
- Keep humans accountable for production decisions.

THE SYSTEM AROUND THE MODEL BECOMES THE ADVANTAGE

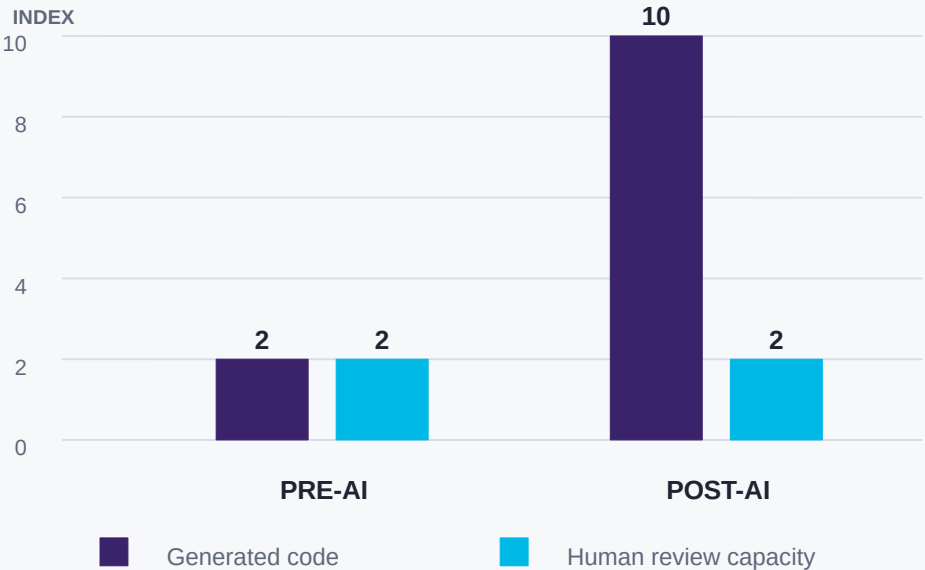
Sources: Google Cloud DORA 2025 · Cory Hymel, V-Bounce Engineering Paradigm

Throughput rises before review capacity catches up

AI can accelerate implementation; the delivery system determines whether that speed becomes value.

ILLUSTRATIVE MODEL · INDEXED VOLUMES

Not survey data. The chart isolates the capacity mismatch.



DELIVERY STABILITY

DORA 2025 reports that AI can improve throughput and product performance while delivery stability depends on the surrounding platform and feedback system.

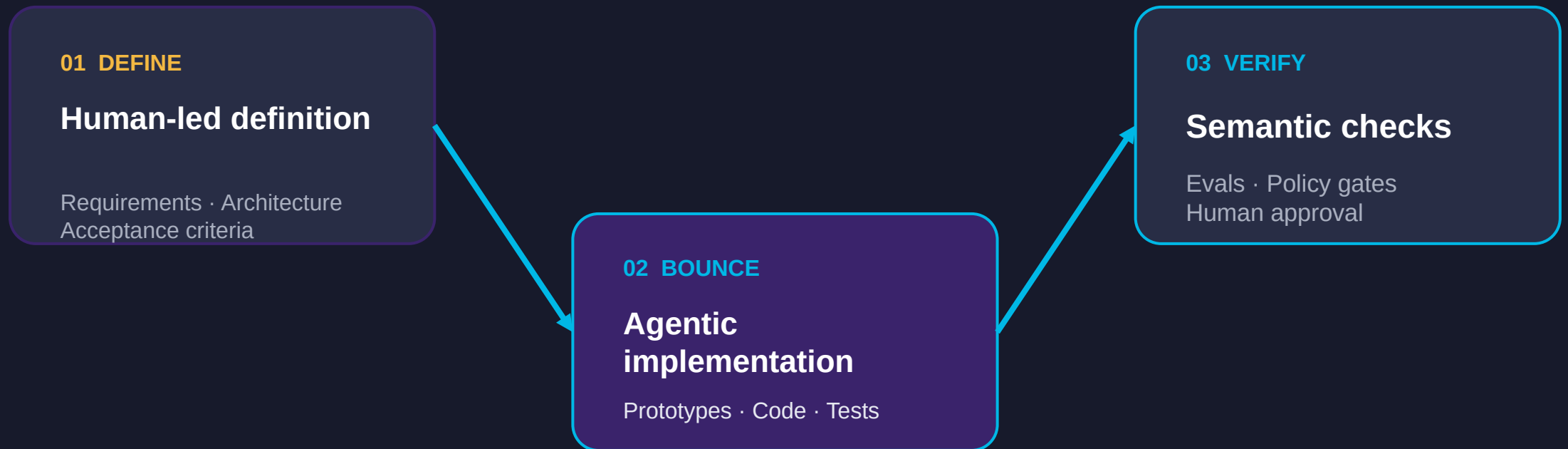
WORKFORCE TRANSITION

Stanford DEL finds stronger pressure on early-career workers in highly AI-exposed occupations. The implication is a changing task mix—not the disappearance of junior engineering.

Sources: Google Cloud DORA 2025 · Stanford Digital Economy Lab 2025

V-Bounce compresses implementation without removing human accountability

HUMAN ACCOUNTABILITY SPANS EVERY STAGE



79%

of Claude Code conversations in Anthropic's cited sample involved some automation—not 79% fully autonomous tasks.

Sources: Cory Hymel, V-Bounce · Anthropic software-development study

Context and evaluation become the engineer's leverage

Reliable output starts with the information, tools and checks surrounding the model.

CONTEXT LAYER

01 Enterprise context
RAG · architecture decisions · policy

02 Design context
tokens · components · structured metadata

03 Tool context
permissions · contracts · observability

THE LEVERAGE ENGINEER

01 Defines the system

02 Orchestrates agents

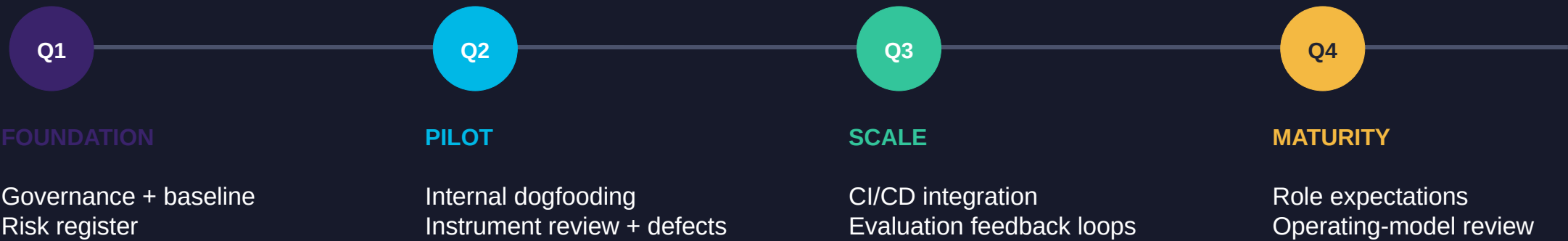
03 Writes evaluations

04 Owns the release decision

The role shifts from writing every implementation detail to designing the conditions for reliable output.

A 12-month pilot turns the model into measurable practice

Sequence governance, learning and integration—then decide what deserves to scale.



PROPOSED PILOT MEASURES

Measure outcomes, not adoption theater.



Roadmap and measures are author-proposed; they are not empirical results.