

Dmitrijs Taranenko

LEAD .NET / CLOUD ARCHITECT | DISTRIBUTED SYSTEMS | AWS

Engineering Evidence: AI-Augmented Delivery

AI is the multiplier. Engineering judgment remains the control plane.

15+ years of hands-on software engineering, now combined with agentic workflows for codebase research, implementation, testing and review. The focus is not faster code generation - it is faster, evidence-driven delivery without giving up architectural ownership or production discipline.

WHAT THE EVIDENCE SHOWS

Research before edits Trace real call/data paths across multiple repositories, identify the smallest safe change surface, then implement.	Root-cause over symptom fixing Use cross-layer investigation and tests to expose transaction, data-integrity and concurrency defects that are easy to miss in local code review.
Verification as a completion gate Repeatedly drive full builds and large automated suites to green, including runs exceeding 500 tests.	Controlled agentic delivery Stop wrong assumptions, reject scope creep and unsafe edits, require explanations, and keep human approval at architecture and risk boundaries.

SELECTED ANONYMIZED ENGINEERING CASES

- **Multi-repository integration analysis** - Mapped an end-to-end workflow across several services and repositories before implementation, validating the actual integration points and avoiding broad speculative changes.
- **Data consistency & concurrency** - Investigated non-obvious persistence failures and surfaced defects involving transaction completion, uniqueness constraints and tenant-level concurrency; validated corrections with automated tests.
- **PR review at delivery speed** - Used a repeatable review-resolution workflow to process batches of unresolved feedback across multiple files, apply targeted code/documentation changes, re-run verification and close the loop.
- **Ticket-to-tested-change workflow** - Combined requirements analysis, codebase research, implementation planning, code changes, tests and documentation into a repeatable end-to-end engineering flow.

OBSERVED WORKING PATTERN FROM CLAUDE CODE INSIGHTS

01 RESEARCH	02 PLAN	03 IMPLEMENT	04 VERIFY	05 REVIEW
----------------	------------	-----------------	--------------	--------------

The recurring pattern is disciplined and auditable: understand the system first, approve the plan, make the smallest justified change, prove it with build/tests, then finish review and documentation.

TECHNICAL CONTEXT

C# / .NET | ASP.NET Core | AWS | Microservices | PostgreSQL | Kubernetes | Docker | CI/CD | GitHub | Claude Code | Agentic Workflows

Best fit: Lead / Principal Backend Engineer, Cloud / Software Architect, or senior distributed-systems roles where hands-on depth, production judgment and modern AI-assisted delivery matter.