ALASKA NORTHSTAR
RESOURCES

GHOST IN THE CODE

Using Agentic AI to Rebuild Understanding of Legacy DAF Systems

Turning opaque mission software into a refactorable asset.

Charlie Jackson, VP of Software Development
August 26, DAFITC 2026

ABOUT THE SPEAKER

Charlie Jackson

VP of Software Development
Alaska Northstar Resources

20+ year career spanning public and private sectors, from writing code to owning a development portfolio. Deep experience understanding and translating how technical decisions land against cost, schedule, and authorization and vice versa.

Federal Software Delivery Leadership

Two decades delivering software for defense and federal civilian missions.

Large-Scale Systems Architecture

Architected and built production systems across on-premises data centers, government cloud, containers, and serverless designs.

Agentic AI in the Development Lifecycle

Leads ANR's adoption of agentic AI across software development, including tooling and engineering practices, and delivery methods.

Technical Governance and Standards

Responsible for ANR's company-wide engineering standards and setting the approval path governing AI use on mission systems.

THE MODERNIZATION TRAP

Too **important** to *ignore*.

Too **fragile** to *change*.

Too **opaque** to *migrate*.

Decades of mission innovation left us systems that are all three at once. The problem is not bad code. It is that the understanding evaporated.

OPACITY IS NOT A DOCUMENTATION PROBLEM

58%+

of modernization engineering is spent on understanding the code, not changing it.

It is a mission risk and a modernization tax

- Every change carries unknown blast radius.
- Every migration estimate is a guess.
- Every cyber finding takes longer to run down.

Citation: Xia et al., *IEEE Transactions on Software Engineering*, 2018; Feitelson, *Communications of the ACM*, 2024.

ECONOMICS OF READING CODE AT SCALE, INVERTED

THEN

- Reverse engineering was linear human labor.
- Priced per engineer-month.
- Understanding did not scale.

NOW

- Breadth-first analysis across millions of lines.
- Cheap, fast, and repeatable.
- The bottleneck moved from reading to judgment.

SINGLE CHATBOT VS. AGENTIC TEAM

SINGLE MODEL

- Passive, one turn at a time.
- You drive every step.
- No memory of the goal.
- Answers questions.

AGENTIC TEAM

- Specialized roles with defined goals.
- The team drives the workflow.
- Artifacts accumulate against acceptance criteria.
- Executes work.

The difference is agency, not intelligence.

SPECIALIZED AGENTS MIRROR MODERNIZATION ROLES

Legacy Analyst
Reads the source and documents it, with every rule cited.

Architect
Designs the target architecture and reviews for alignment.

Migrator
Builds the modernized system to the requirements.

Test Engineer
Proves output parity, deliberately separate from the builder.

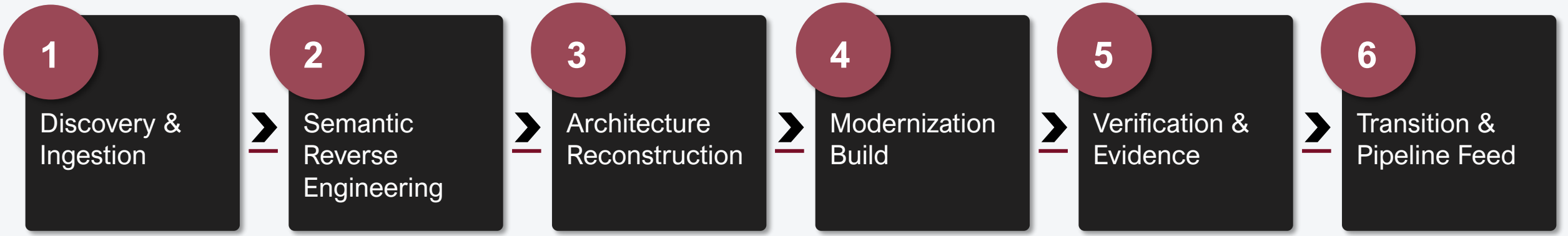
DevSecOps Engineer
Wires the CI/CD pipeline with blocking security scans.

Compliance Officer
Produces validated NIST 800-53 control evidence.

Each role has a narrow job and a definition of done.

CODE MODERNIZATION IN SIX PHASES

From **millions of lines** of code to an organized modernization pipeline, in **six phases**:

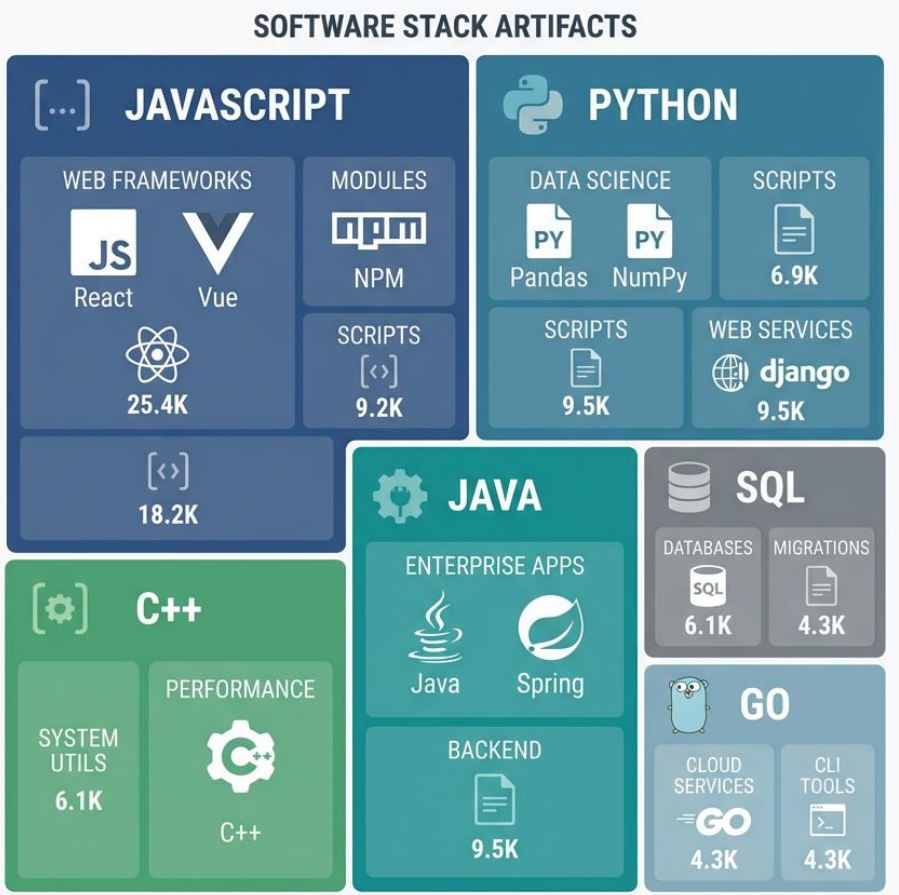


Human gate reviews sit between every phase.

PHASE 1 | DISCOVERY

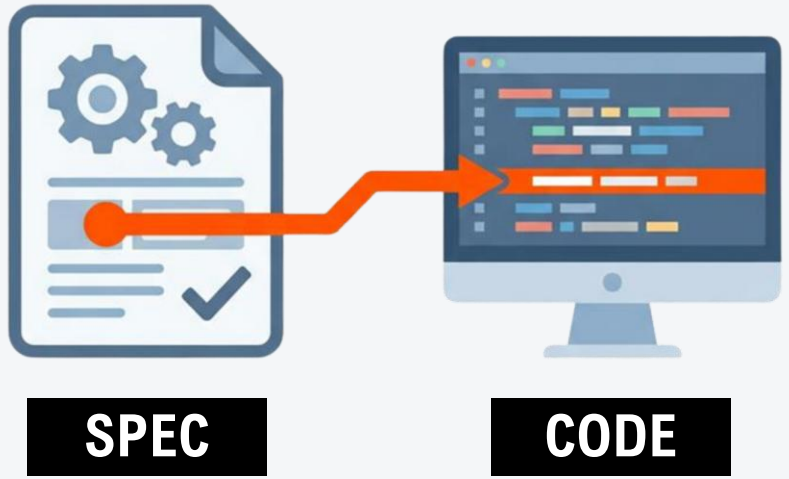
You cannot analyze what you have not counted.

Agents crawl the whole mixed-language estate, source, copybooks, job control, scripts, config, and build files, and produce a complete inventory of what exists and what is missing.



PHASE 2 | SEMANTIC REVERSE ENGINEERING

No rule is invented. Every rule is traceable.



Agents infer architecture, data flows, control logic, and business rules. Each inferred rule is cited back to the exact legacy source file and line it came from.

PHASE 2 | SEMANTIC REVERSE ENGINEERING

The estate becomes sequence diagrams, data dictionaries, interface catalogs, and dependency graphs:

Sequence Diagrams

How the system actually behaves, step by step.

Data Dictionaries

Every field and structure, defined from the source.

Interface Definitions

The catalog of what connects to what.

Dependency Graphs

The real map of how components rely on each other.

Generated from the code, regenerated when the code changes.

PHASE 2 | SEMANTIC REVERSE ENGINEERING

The same pass exposes duplicate logic, dead code, brittle integrations, and security anti-patterns:

Duplicate Logic
The same rule implemented many times, inconsistently.

Security anti-patterns
Hard-coded secrets, weak crypto, unsafe input handling.

Brittle Integrations
Fragile couplings that break under change.

Dead Code
Paths that no longer execute but still carry risk.

Found systematically across the whole estate, not anecdotally.

PHASE 3 | ARCHITECTURE RECONSTRUCTION

Related components get grouped into functional domains that map to mission capability

BEFORE

- A tangled dependency hairball.
- No natural seams to modernize along.
- Every change risks the whole system.

AFTER

- Clean clusters labeled by mission function.
- Modernize one domain at a time.
- Domains, not files, become the unit of work.

Each domain carries a debt-and-risk score, so leadership can sequence modernization highest-risk first.

PHASE 4 | MODERNIZATION BUILD

The modernized system is built one domain at a time, from requirements traced to the source:

Build to Spec
Every requirement carries its citation, so the Migrator builds to the traced rule, not to guesswork.

One Domain at a Time
Work follows the seams found in reconstruction, so a change never destabilizes the whole estate.

Builder Does Not Grade Itself
The Migrator writes the code. A separate agent proves it. Correctness is decided in Phase 5.

The build follows the blueprint, not a guess.

PHASE 5 | VERIFICATION & EVIDENCE

The modernized code is proven against a verified golden baseline, field by field:

Seed the Scenarios
Cover every significant branch: boundary values, error paths, and negative cases.

›

Freeze the Golden Baseline
Capture the legacy system's actual outputs as an independent, verified reference.

›

Diff for Parity
Compare modernized outputs field by field. Done means matched, not reviewed.

Parity is measured, not asserted.

PHASE 6 | TRANSITION & PIPELINE FEED

The artifacts flow straight into backlogs, CI/CD, and model-based engineering:

Prioritized Backlog

Domain models and risk rankings become the agile backlog.

CI/CD Pipeline

Wired with security scans that block on findings.

Infrastructure as Code

Deployment environments defined and repeatable.

Evidence Package

Control evidence compiled alongside the build.

The output of understanding is the input to modernization.

WHERE THIS WORKS...AND WHERE IT DOES NOT

AI wins on breadth, pattern detection, dependency mapping, and debt discovery

Reads everything
The whole estate, without fatigue or sampling.

Detects patterns
Finds what a human spot-check would miss.

Maps dependencies
Exhaustively, not anecdotally.

Ranks debt
Objectively, across the entire code base.

Any task that rewards reading everything, the machine does better.

WHERE HUMANS STAY CENTRAL

The code tells you what it does, not why it must.

Mission Intent

Which behaviors are load-bearing for the mission.

Risk Tradeoffs

Which risks are acceptable and which are not.

Operational Edge Cases

The real-world context no one wrote down.

LESS ARCHAEOLOGY & MORE JUDGEMENT

This removes the archaeology that precedes every modernization, and returns engineers to judgment:

BEFORE

AFTER

Same people, higher-value work.

TRUST THROUGH TRIPLE-LAYER VERIFICATION

No single check is trusted, so verification runs in three independent layers:

LEVEL 1

Executable Checks

Gate scripts and the full test suite.

Catches build breaks and functional regressions.

Limit confirms the tests pass, not that the behavior is right.

LEVEL 2

Adversarial Agent Review

A separate agent, such as the Architect, reviews work it did not build.

Catches design drift and architecture the tests never assert.

Limit still an agent, so prose reviewing prose is not proof alone.

LEVEL 3

Human Review

A person validates the checks and sanity-checks the agent.

Catches mission-intent errors and results that pass but are wrong.

Limit reserved for judgment, not re-reading every line.

Each layer catches what the one before misses. The human validates the checks, not the code.

WRITE. TEST. PASS. COMPLETE.

The agent that writes the code is not done until the tests pass in its own session

The Builder Must Run the Build

No declaring victory on unexecuted code. The full test suite must be green before completion.

Pin the Environment First

The toolchain is pinned and smoke-tested up front, so environment problems never masquerade as quality problems.

Cheap retries beat heroic recovery: a failure costs one phase, not the engagement.

GOVERNANCE

Every engagement produces its own traceability, provenance, and decision record automatically:

TRACEABILITY

Gate Ledger

Records what was verified, and when.

ACCOUNTABILITY

Decisions Log

Records which ambiguity a human resolved, and when.

REPRODUCIBILITY

Configuration Pin

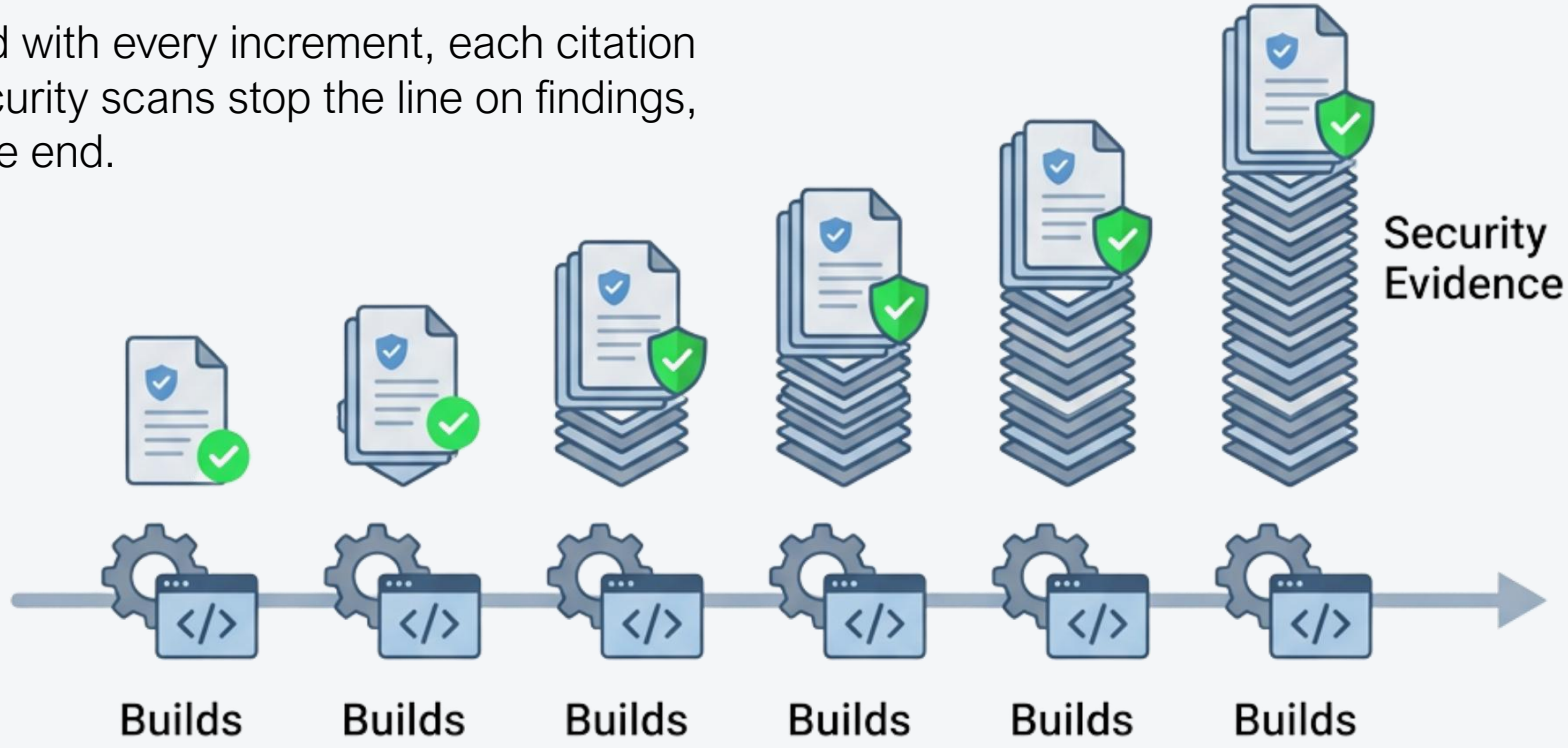
Records the exact tooling version used.

You can always answer what ran, what was checked, and who decided: the record that makes RMF and AI governance auditable.

CYBERSECURITY & ATO

The body of evidence is a build output.

NIST 800-53 control evidence is generated with every increment, each citation validated against the catalog. Blocking security scans stop the line on findings, so ATO preparation is not a scramble at the end.



CLOUD-NATIVE & ZERO-TRUST READY

Understand once, refactor toward whichever target the mission needs.

Containers

Lift understood domains into portable services.

Microservices

Decompose along the seams the analysis found.

Serverless

Map event-driven functions to the domain model.

START WITH ASSESSMENT & END WITH PROOF

DO NOW

1. **Conduct a bounded assessment on one representative, well-scoped legacy system.**

REQUIRE OF ANY PROVIDER

2. Executable verification gates as acceptance criteria, not status-meeting assurances.
3. Output parity against a verified golden baseline as a contractual acceptance criterion.
4. Your SMEs' time reserved for the mission-intent and risk calls only they can make.
5. Traceability and control-evidence records delivered as artifacts, not promised.

And when you evaluate tools: judge them on verification discipline, not agent count.

TURNING UNCERTAINTY INTO A PLAN

Living Documentation

An understood system, generated from the source.

Ranked Debt Hot Spots

A risk-ordered view of where to act first.

Target-Agnostic Domain Model

Ready for containers, microservices, or serverless.

Directional Approach

A defensible recommendation for what comes next.

Low commitment, high clarity.

FROM GHOST TO BLUEPRINT

The goal is not AI that replaces engineers, it is a clear blueprint for next-generation DAF capability.

Turn the ghost in the code into the foundation for what comes next.

Questions?

Charlie Jackson, VP of Software Development
August 26, DAFITC 2026
charlie.jackson@aknorthstar.com