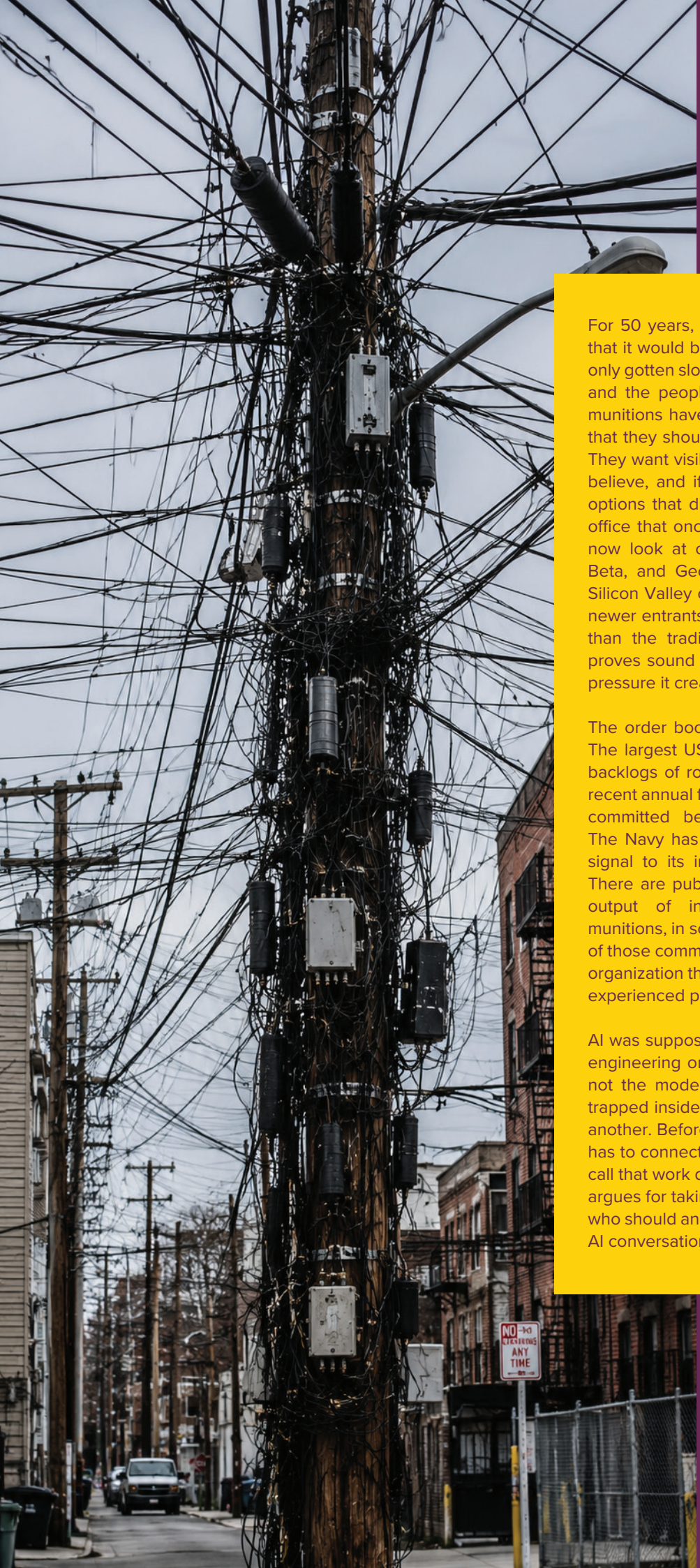
The Engineering Systems Problem Behind Failed AI Programs

Why the AI conversation in aerospace and defense skipped a generation of engineering work

Author:
Chris Schwartzbauer,
CEO, Quest Defense



For 50 years, this industry has promised its customers that it would build things faster, and for 50 years, it has only gotten slower. Delivery dates stretch, programs slip, and the people who buy aircraft, ships, engines, and munitions have stopped accepting the old reassurance that they should go away and check back in a decade. They want visible progress now, in increments they can believe, and if a prime cannot show it, the buyer has options that did not exist a few years ago. A program office that once chose among a handful of names can now look at companies like Anduril, Palantir, Archer, Beta, and Gecko Robotics that ship technology at a Silicon Valley cadence, and conclude that one of these newer entrants may deliver a working capability sooner than the traditional primes. Whether that conclusion proves sound is a separate argument. The competitive pressure it creates is already here.

The order books show how concrete that pressure is. The largest US defense primes are carrying combined backlogs of roughly half a trillion dollars on their most recent annual filings, several years of production already committed before a single new award is signed. The Navy has issued the largest shipbuilding demand signal to its industrial base since the Roosevelt era. There are public commitments to sharply increase the output of interceptors and long-range precision munitions, in some lines by a factor of two or more. Each of those commitments comes due inside an engineering organization that has to move faster precisely as its most experienced people retire.

AI was supposed to take some of this weight. For most engineering organizations it has not, and the reason is not the models. The data the models would need is trapped inside tools that were never built to talk to one another. Before a single use case can deliver, someone has to connect the systems and make the data usable. I call that work digital plumbing, and the rest of this paper argues for taking it seriously and for being honest about who should and probably could do it. Only then does the AI conversation fall into a sensible order.

The AI conversation skipped a generation of work

Most of the AI use cases pitched into aerospace and defense assume a level of data access that may not exist on the floor. The pitch could describe a model that reads requirements, simulation results, test data, and field telemetry, reasons across all of it, and hands an engineer a better answer. The model may be able to do that. However, the data cannot get to it because it lives in formats and repositories chosen one tool at a time by different functions, for different reasons, with no expectation that anything downstream would ever need to read them.

The share of companies abandoning most of their AI initiatives rose from 17% to 42% in a single year, and only a small fraction of enterprise pilots ever reach production. Gartner expects more than 40% of agentic AI projects to be canceled by the end of 2027, citing escalating costs, unclear business value, and weak risk controls. The models worked; what failed was everything around them, the connections and the data. Many AI use cases have ended in pilot purgatory, the state where a demo works well enough that no one will kill it, and the plumbing is broken enough that no one can scale it.

I am not saying that AI isn't working. In fact, these false starts point to a recurring lesson. The plumbing comes first, and the AI depends on it. When an organization attempts AI on broken plumbing, the model trains once on a hand-assembled data set, demonstrates beautifully, and then cannot be retrained or redeployed because the pipeline that fed it was a person exporting files. The work that would have made the use case real was the unglamorous connection work that nobody funded.

What digital plumbing actually means

Inside a typical platform program the engineering tool estate has grown into something like a small city built without zoning. Requirements live in DOORS, design in CATIA, NX, or Creo, system models in Cameo or Rhapsody, analysis in MATLAB, Simulink, and ANSYS, product data in Teamcenter, Windchill, or 3DEXPERIENCE, and test and verification in stacks of their own. Each tool is competent at its job. The trouble is the spaces between them, where a requirement has to become a model parameter, a simulation result has to inform a design change, and a test outcome has to close a verification item. Those handoffs are where the work usually gets lost.

Traceability from a stakeholder need down through the system model, the interface definitions, the geometry, the bill of materials, and the test cases breaks somewhere in the chain, so no one can answer a simple question about what a given change touches.

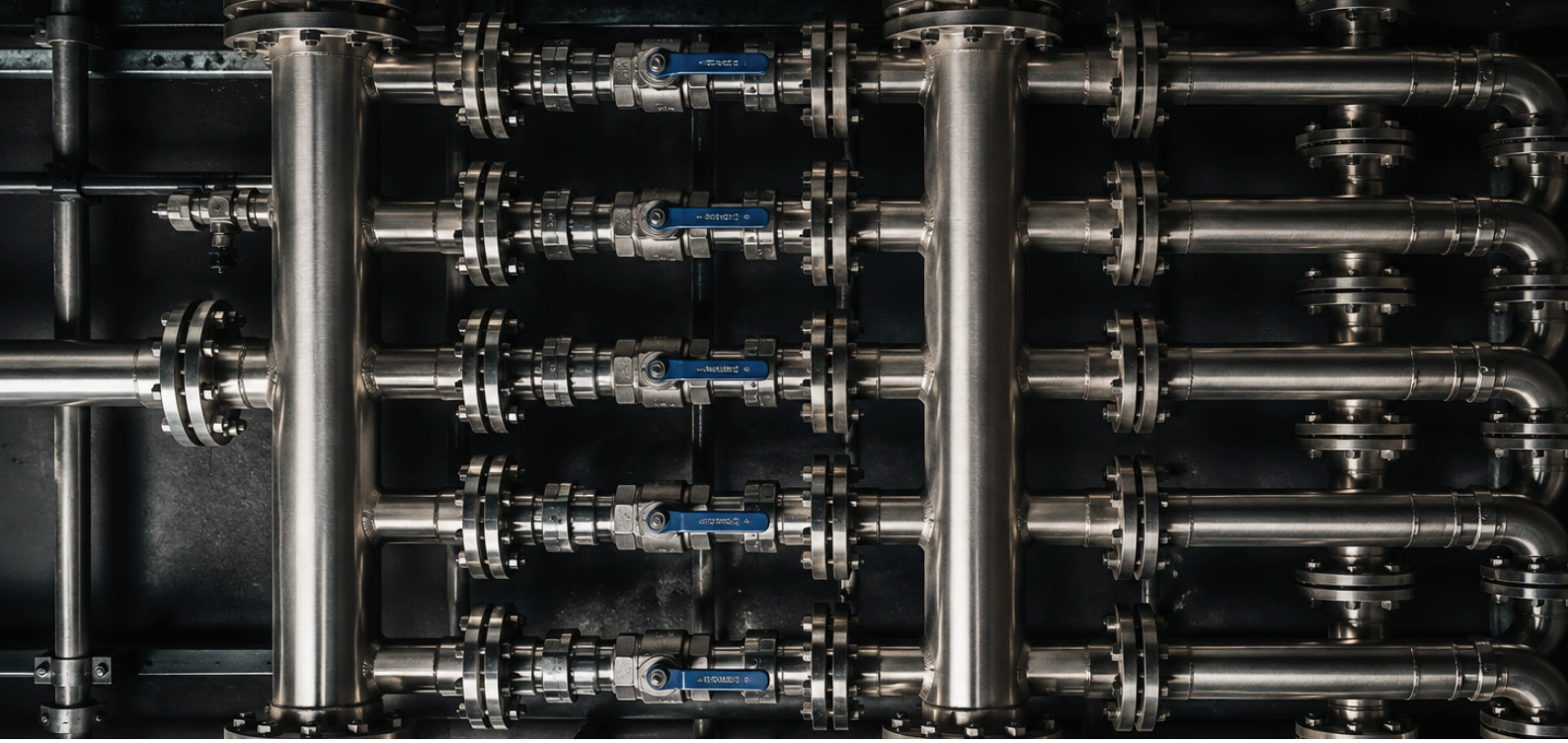
Verification status has no single dashboard, so closure gets tracked in spreadsheets. Interface mismatches surface late, as interference reports and engineering change orders that arrive long after they should have. It is the direct cost of systems that cannot read each other.

Significant pools of tool-specific data stay unusable for anything upstream or downstream, and the value locked inside that data cannot move. Nobody needs to tear the house down and rebuild it. Putting in one bathroom that works is enough, getting hot water to the place that needs it, connecting the two systems that have to talk, proving the data can flow, and letting the organization decide what to do with the lesson.

Why this is an engineering problem

The easy explanation is that this is an IT failure, that someone in a central function picked the wrong platforms and never integrated them. The less comfortable truth is that tool sprawl is the residue of engineers doing their jobs. An engineer with a deadline and a gap in the toolchain does what good engineers have always done, which is build something that closes the gap. The bespoke script, the custom interface, the one-off converter, all of it exists because necessity demanded it, and nobody had a better option in that moment.

The engineer's instinct collides with how IT approaches the same problem. IT wants to buy a tool, standardize on it, and support it for years. The engineer wants to solve the specific problem in front of him and does not much care whether the solution generalizes. Both are rational, and neither wins on its own, which is why the problem has lasted. The custom tool rots because keeping it current was never anybody's job. The off-the-shelf tool is general, where the work is specific, leaving the engineer the exact gap he built the bespoke thing to close. The same fault line runs through the AI question, where IT wants to procure a platform, and the engineer says he does not have a platform problem; he has an engineering problem. This is closer to a clash of cultures, even of belief, and no purchase order resolves it.



Two clocks that do not keep the same time

The culture clash hides a difference in objective that almost no one states out loud. The IT or digital transformation organization is trying to build a sustainable, ubiquitous architecture, a system meant to serve the whole organization more or less forever. Enterprise ERP is a reference case. Its integrations run for 20 years and are never quite finished, because the goal is permanence and the target keeps moving. The engineering organization has a different and more immediate mandate, which is to get a product built and out the door, because that is how the business makes money. One side is building utopia. The other has to make payroll. Both objectives are legitimate, and they run on clocks that do not keep the same time.

Why an OEM cannot solve this from the inside

Two clocks that cannot be reset from inside one organization raise an obvious question, which is who reconciles them. The answer points outside, toward engineering-centric companies that span the full cycle and, by definition, have to keep proving their value to stay relevant. An aerospace and defense OEM is organized around deep subsystem specialties, and it has to be. The avionics group, the hot-section team, the engine-controller people, the stress analysts, the verification organization, each needs profound knowledge of the piece they own, and that depth is the whole point. What the structure does not produce is the broad view across the program. Each group is right about its own domain, and none owns the connections between domains, which is exactly where the data has to flow.

A second and underestimated reason is that a subsystem team does this integration work only once. They build it for their own system and then move on. In my personal life, I have sweated copper pipe, heated the joint with a torch, and connected two lines, and it is as much art as science. The joint held, but it was not pretty, and I would not bet on it lasting 10 years. A plumber does that joint in two minutes, and it holds, because the plumber does it every single day. A company that does integration across many programs is the plumber, setting up the systems that let customers run them every day, often enough that the joints hold.

The way the work fractures inside an OEM makes this worse. Design, integration, validation, and production are split into silos and pass work down a serial line, so the teams that most need to communicate often do not. Testing and verification get treated as an afterthought, which is exactly backward, because the verification stack is frequently the thing that determines whether anything else can close.

Why the tool vendors were never going to fix it

If the OEM cannot close these gaps from inside, the next assumption is that the platform vendors will. Siemens, Dassault, and PTC have not solved interoperability in 30 years, and the reason is in how the incentives run inside a software company. You are rewarded for building a more complete package and for giving the customer one more reason to stay inside your stack. Opening clean interfaces or exposing APIs only invites substitution, while a closed, complete suite makes you hard to rip out. Faced with that choice, a rational software business builds the moat.

Engineering organizations live with the result. Proprietary formats and schemas make exchange hard, and custom integrations are expensive to maintain. Open standards exist, OSLC for linking lifecycle data, STEP for exchanging product models, SysML for system models, precisely because the commercial incentive runs the other way and someone had to push against it. Expecting vendors who profit from depth in their own stack to volunteer the connective work is expecting them to act against their own interest, which is why the layer has stayed broken.

Core versus critical, and what an OEM should actually own

If neither the OEM nor the vendors will fix the plumbing, the real question is what the OEM should own. The resistance to answering it usually comes back as a single sentence. We need to be the experts. The right response is a question, expert at what? You need to be the systems engineering experts who know how to design the product, how it should behave, and what good looks like when it is finished. The tool that plugs your model into your development chain is not that, and the build environment is not the product. A general contractor builds the house and sells it, and is accountable for the house, but does not personally wire the outlets or run the supply lines, and is no worse at building houses for having hired an electrician.

MBSE is where this gets genuinely confused, because the model and the tooling get treated as the same thing. The model is core. It is your design intent made executable, and it should live with your best engineers. The environment that connects that model to everything around it is plumbing, and treating the plumbing as a core competency is how it ends up owned by everyone and mastered by no one.

Underneath the organizational case is a question of capital, because the engineering talent to do this work is the scarcest resource the industry has. The 2025 AIA and McKinsey workforce study found 76% of companies struggling to hire engineering talent and industry attrition running near 15%, more than double the rate across other sectors. Close to 29% of the aerospace and defense workforce is over 55, and by Deloitte's count a quarter of it carries more than 20 years of experience and is at or past retirement age. The clearance pipeline makes it tighter still, with the number of Americans holding a security clearance down by close to a million since 2013 while demand keeps climbing, which leaves tens of thousands more cleared positions open than there are people to fill them. Using your most experienced engineers on bespoke tool integrations, when the same people could be on the product only they can build, fails the most basic test of stewardship.

The transformation roadmap still gets signed, to the wrong tune

The big transformation program tends to ignore that same engineering reality. A fashionable claim holds that the program is dead, that no one signs a 5-year roadmap anymore. My experience runs the other way. The wholesale roadmap still gets signed, with IT organizations and consulting firms, built around new tools, new processes, and data structuring. Those programs are tuned for standardization. The engineering reality they meet is different, defined by the demand for higher production rates, the attrition of the senior experts who hold the know-how, and the shortcomings in the supply chain.

The work that survives contact with an engineering floor is incremental and leverageable. You prove value inside a quarter, you keep what works, and you let the lessons feed the larger transformation or not. The market forces this. A buyer who has heard faster promised for 50 years wants to see progress on the board, because the promise alone has lost its value. A wholesale approach can still make sense, as long as someone validates it against the engineering reality. If the program is run by partners fluent in data structures but who have never had to get production rates up against an aging workforce, it will produce a clean architecture and miss the point.

AI, put back in its place

All of this puts AI back where it belongs. Using it is table stakes now, expected of anyone serious, and the question was never who is allowed to do the work, whatever some vendors claim about earning the right to it. The real question is which use cases are real, and that is what the plumbing answers. What it buys you is sight, the ability to see what AI can and cannot do in a specific environment. Whoever connects the systems and watches what moves through them every day learns what data actually exists, what the pipe can carry, and which proposed use cases are real and which assume access that was never built. Take the integration seriously and the AI conversation changes shape. The fantasy use cases fall away because their data was never reachable, and what is left are the ones you can actually deploy and keep retraining in production, because the pipeline feeding them is real instead of a person exporting files at midnight. The plumbing is what lets you tell the difference between a use case and a wish.

How to start, the smallest plumbing that pays

The way into this work is small and specific. Pick one integration tied to a known cycle-time pain, simulation results flowing into product data, or test outcomes into a dashboard that did not exist before. Scope it to 8 to 12 weeks. Prove the data moves and a real bottleneck eases, keep the lessons, and then decide, with evidence in hand, whether to fold the approach into something larger.

Funding is the part that defeats most plumbing projects, because the work is invisible to the people who hold the budget. Integration on its own line item loses every contest with a project that has a demo and a story. The way through is to attach the plumbing to an outcome the chief executive already cares about, a production rate or a program cost the board is watching, and to present the integration as the thing that makes that outcome reachable. The plumbing is the enabler. It should almost never be sold as the deliverable.

The last piece is political, and it explains why an outside partner can move where an internal team cannot. Each function owns its tools and treats the seams between them as someone else's problem. The CIO has the standardization mandate but lacks engineering credibility with the design teams, and the chief engineer has the credibility but not the IT remit. A partner with no stake in the internal turf can hold both objectives in the same room and broker between them precisely because it is not trying to win the territory.

The expensive instinct

The instinct that resists all of this is the instinct to keep the work inside the house, and it is the expensive one. It treats the build environment as if it were the product, and it spends scarce senior engineering time and real AI budget on a foundation that was never built, a bill that grows every year the talent gets scarcer and the schedules get tighter.

The product is yours. It should be, and it always will be. The plumbing does not have to be. Connect the systems, make the data usable, and the harder conversations about AI, autonomy, and digital engineering become possible to have honestly. Leave the plumbing broken, and those conversations stay exactly what they are today. They never become capabilities.

Sources:

1. Combined order backlogs compiled from the most recent annual filings of the largest US defense prime contractors (fiscal 2025). “Throughput, not just innovation, may define the future of US defense manufacturing and industrial scale,” 2026, at https://www.deloitte.com/us/en/insights/industry/aerospace-defense/us-defense-manufacturing-industrial-scale.html.
2. White House Office of Management and Budget, fiscal 2027 budget proposal, 2026. The OMB statement describes the nation's 41-ship shipbuilding order as the largest demand signal to the maritime industrial base since the administration of Franklin D. Roosevelt. Reported in Stars and Stripes, https://www.stripes.com/branches/navy/2026-04-03/pentagon-spending-plan-41-new-ships-21271576.html.
3. Sources for the adoption and abandonment figures in this passage. S&P Global Market Intelligence found the share of companies abandoning most of their AI initiatives rose from 17% to 42% year over year (https://www.spglobal.com/market-intelligence/en/news-insights/research/2025/10/generative-ai-shows-rapid-growth-but-yields-mixed-results). IDC's CIO Playbook 2025, commissioned by Lenovo, found that for every 33 AI proofs of concept an organization starts, roughly 4 reach production (https://www.insight.com/en_US/content-and-resources/brands/lenovo/idc-cio-playbook-2025-its-time-for-ai-nomics.html).

   MIT Project NANDA's The GenAI Divide reported that roughly 95% of enterprise generative AI pilots produced no measurable impact and located the cause in integration and data readiness (https://fortune.com/2025/08/18/mit-report-95-percent-generative-ai-pilots-at-companies-failing-cfo/). Gartner predicted in June 2025 that more than 40% of agentic AI projects will be canceled by the end of 2027, citing escalating costs, unclear business value, and inadequate risk controls. (https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027). For a 2026 analysis tracing those cancellations to deployment and data access, see Forbes, https://www.forbes.com/sites/robertszczerba/2026/07/07/why-40-of-agentic-ai-projects-may-be-canceled-by-2027/.
4. Aerospace Industries Association and McKinsey, “Accelerating progress: Maximizing the return on talent in A&D,” 2025, drawing on more than 30 A&D organizations representing over 600,000 US employees. The study reports that 76% of member organizations face sustained challenges hiring engineering talent and that industry attrition holds near 15%, more than double the cross-industry average. See https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/accelerating-progress-maximizing-the-return-on-talent-in-a-and-d.
5. Workforce age share of 29% over 55 from AIA and PwC (2023 estimate). The figure that 25% of the workforce carries more than 20 years of experience and is at or beyond eligible retirement age is from Deloitte, “2025 Aerospace and Defense Industry Outlook,” at https://www.deloitte.com/us/en/insights/industry/aerospace-defense/aerospace-and-defense-industry-outlook/2025.html.
6. The number of Americans holding a security clearance has declined by close to one million since 2013, while demand for cleared specialists has continued to rise. Industry recruiting data points to roughly 70,000 more cleared positions than there are available cleared professionals to fill them. See ClearanceJobs, https://about.clearancejobs.com/employers/learn-more/hiring-trends.

For further information or queries, please reach out to us at information@quest-defense.com