ADAMBOAS.COM • PAPER • September 11, 2026

The Next Software Bottleneck Is the Capacity to Absorb Change

DoWI 8430.01 is published. The next step is making acceleration usable.

Adam Boas

Published September 11, 2026

Contents

Executive Summary

In 2024, I drafted the original policy that became DoWI 8430.01, *Accelerated Mission Software*. It took effect on September 8, 2026. I am proud of that work, and grateful to the people who refined, coordinated, and carried it to publication.

The instruction matters because it treats software as an enduring mission capability. Delivery, security, testing, governance, and sustainment belong to the same operating problem. Publication is a milestone. It is also the beginning of the next argument.

As AI expands the volume of work that software teams can produce, the Department must expand its capacity to verify, accept, field, and sustain that work. Otherwise, faster production will create larger queues for the same constrained reviewers, testers, authorizing officials, and mission owners.

The next software bottleneck is the capacity to absorb change.

1 Publication Is the Starting Line

DoWI 8430.01 establishes direction. It treats software as an enduring mission capability and connects delivery, security, testing, governance, and sustainment across the lifecycle. The next work is to turn that direction into repeatable behavior across the Department's software enterprise.

That work requires more than acquiring AI tools. It requires deciding what work software actors may receive, making their boundaries enforceable, producing evidence that survives handoffs, and preserving human judgment where it is consequential.

I wrote the original policy to help the Department deliver software as an enduring mission capability. The work I am advancing now addresses what happens when the producer of that capability changes, and when the volume of proposed change exceeds the institution's capacity to process it.

2 What the Instruction Makes Possible

The instruction provides a foundation for addressing that problem. It directs teams to optimize flow, integrate security and quality, automate continuously, and adapt governance to iterative delivery. It measures the complete value stream through operational use, not simply the performance of an isolated development step. Those are connected requirements, not separate modernization initiatives. DoWI 8430.01, Sections 3.2–3.3.

The enterprise digital arsenal extends that logic across organizational boundaries. Platforms and factories must produce and share machine-readable evidence; authorizing officials must accept standard evidence to grant reciprocity and avoid redundant assessments. Software assurance becomes an evidence-producing function of delivery, with build provenance, component transparency, application security, and continuous testing supporting decisions throughout the lifecycle. DoWI 8430.01, Sections 3.4–3.5.

The AI provisions carry the same discipline forward. AI-generated code is unverified input. Developers remain accountable. Security- or safety-critical changes require human review and approval. Approved environments, data protections, rigorous validation, and AI-component traceability remain

part of the operating baseline. DoWI 8430.01, Section 3.6.

My reading is straightforward: speed should come from engineering assurance and accountability into delivery, rather than repeatedly reconstructing them around it.

The next step is to make that expectation work when software can produce proposed changes faster than people can individually examine them.

3 Faster Production Can Create Slower Delivery

Consider an agent that clears a dependency-remediation backlog by generating updates, tests, and pull requests. The demonstration looks successful. More work has been produced in less time.

But someone must still establish whether the changes preserve required behavior, whether the tests are adequate, whether security assumptions remain valid, and whether the resulting software is ready for its intended environment. If those decisions require the same manual reconstruction for every change, the agent has accelerated arrival into the queue without increasing its throughput.

The backlog moved.

This is not inevitable. AI can help generate evidence, identify affected requirements, and prepare review material as well as write code. But those functions have to be designed together. Adding another agent to review the first agent does not automatically create independent assurance, and a persuasive explanation is not proof that a requirement was satisfied.

Nor is every program primarily constrained by review. Funding, architecture, integration access, and operational testing can remain decisive. The point is to locate the actual constraint and improve the complete path to mission use.

The distinction is between **proposed output and accepted work**. A completed task is not necessarily a deployable change. A deployable change is not necessarily demonstrated mission value.

If accepted throughput rises while total burden and defects remain controlled, the demonstration has produced evidence of useful capacity. If the review queue grows or rework consumes the gain, it has identified the next engineering problem.

Both findings are more useful than a count of agents deployed or pull requests opened.

This is the distinction developed in *The Agentic Information Enterprise*: workforce augmentation becomes credible when the institution can assign dependable additional work without transferring equal or greater burden to its people. Non-person entity identity provides technical accountability. It does not confer public office or independent governmental authority.

4 Apply the Same Discipline to Policy Implementation

The policy office belongs in this demonstration too.

A new instruction creates work across Component guidance, reference designs, contract requirements, platform controls, training, and evidence expectations. A bounded agent could trace selected obligations to existing implementation artifacts, identify gaps and conflicts, and prepare source-linked changes for review.

It could not resolve an authoritative interpretation, issue policy, grant a waiver, or accept governmental risk. Those decisions remain with the responsible officials. Its value would be in making their work more complete, traceable, and timely.

This is the implementation connection to *From PDFs to Pull Requests*. An issued instruction remains authoritative. Its implementation can become versioned, reviewable, testable, and connected to operational evidence. A failed control or conflicting requirement can then lead to a specific correction rather than another unstructured document cycle.

That feedback also matters for restraint. Implementation evidence should help distinguish a missing policy requirement from a requirement that has not yet been made operational.

5 Connect Authority, Evidence, and Acceptance

Acceptance is the set of decisions that makes transitions from proposed output to deployable change and accepted mission work legitimate. It should be designed into the workflow, not installed at the end as another board.

My recent writing approaches this problem through three complementary functions. The Continuous Assurance Fabric Reference Architecture (CAF-RA) and Agent Control Plane Reference Architecture (ACP-RA) are theoretical design proposals, not Department-adopted standards. *The Agentic Information Enterprise* adds a proposed operating model for assigning, supervising, and measuring non-person agent work.

The point is not to adopt the acronyms. It is to connect the functions.

5.1 Authority Defines What May Be Attempted

An agent needs a distinct non-person entity identity, scoped credentials, approved tools and data, an accountable owner, and explicit limits. Controls must enforce those limits at the point of action, outside the agent's discretion. A prompt may describe the work; it cannot grant permissions. The system must be able to stop the work, revoke access, and contain its effects.

5.2 Evidence Establishes What Happened and What Remains Supported

A test result must resolve to the artifact, configuration, environment, and requirement it actually evaluated. Evidence needs provenance, validity conditions, and a path to verified remediation. The producing agent should not be able to redefine success or bypass the checks used to establish it. Changes to models, tools, permissions, and dependencies must trigger reassessment where they affect the claim.

5.3 Acceptance Determines Whether the Result Closes Useful Work

Before execution, define the intended outcome, required evidence, accepting authority, and exception path. The relevant humans retain decisions reserved to them. Bounded, non-reserved effects may be candidates for explicitly preauthorized automation under applicable rules; that is not permission for an agent to authorize itself.

These functions should use existing identity, delivery, evidence, and authorization services wherever possible. Their value comes from the relationships between them, not from creating another enterprise platform.

Reciprocity makes those relationships especially important. A receiving team needs to know what has already been established and what remains specific to its mission. Build integrity does not, by itself, demonstrate operational effectiveness. Evidence can travel while the conditions supporting a

particular claim change. Independent government testing remains necessary under the instruction. DoWI 8430.01, Sections 3.4.d and 3.5.d.

The objective is to eliminate redundant reconstruction while preserving consequential judgment.

6 Prove Capacity, Not Activity

The practical next step is a bounded implementation in an existing, appropriately approved delivery environment.

Start with a defined class of dependency remediation and regression testing. Establish acceptance criteria, current cycle time, reviewer effort, rework, and quality before introducing agents. Keep merge and release decisions with designated humans during the initial demonstration.

Have agents prepare changes and supporting evidence. Verify the work through separately controlled checks. Exercise failure conditions: a malicious repository instruction, revoked credentials, missing evidence, an attempted scope violation, and a change that passes automated tests but fails a mission-relevant requirement.

Then ask a second authorized team to evaluate which evidence it can reuse and which claims require additional work in its environment.

Measure the result in accepted work, with review capacity held comparable and quality thresholds established in advance. Count supervision, correction, assurance, platform operations, sustainment, and incident burden. Distinguish newly completed backlog from human time actually released on comparable tasks.

7 Conclusion

Publication of DoWI 8430.01 makes acceleration possible. Implementation must make that acceleration usable.

The next measure of software modernization is how much useful change the Department can absorb, not how much output its tools can generate.

Author's analysis and proposed implementation approach. DoWI 8430.01 is the authoritative policy; the architectures and operating concepts referenced above are independent proposals.

8 References

1. Department of War, DoWI 8430.01, *Accelerated Mission Software*, September 8, 2026.
2. Adam Boas, *The Agentic Information Enterprise*.
3. Adam Boas, *Continuous Assurance Fabric Reference Architecture (CAF-RA)*.
4. Adam Boas, *Agent Control Plane Reference Architecture (ACP-RA)*.
5. Adam Boas, *From PDFs to Pull Requests*.