SYM-AM-25-340

Excerpt from the Proceedings

of the

Twenty-Second Annual Acquisition Research Symposium and Innovation Summit

Thursday, May 8, 2025 Sessions
Volume II

A New Path Forward: An Analysis of Current AI Software Acquisition Procedures

Published: May 5, 2025

The research presented in this report was supported by the Acquisition Research Program at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

ACQUISITION RESEARCH PROGRAM
DEPARTMENT OF DEFENSE MANAGEMENT
NAVAL POSTGRADUATE SCHOOL

A New Path Forward: An Analysis of Current AI Software Acquisition Procedures

Richard Beutel—Greg and Camille Baroni Center for Government Contracting, George Mason University. [rbeutel@gmu.edu]

Abstract

Maintaining a competitive edge in AI enabled software requires sustained investment in research, development, and workforce training. But it also requires an evaluation of the specific acquisition tradecraft applicable to the procurement and deployment of transformative software technology.

Currently, contracts used for DoD software development programs, including software containing embedded AI elements, are negotiated using a hodgepodge of existing contract vehicles, accelerated procurement frameworks and acquisition tradecraft approaches.

While completely reinventing the wheel may be a bridge too far, we believe it helpful to examine the current status quo pertaining to software acquisition procedures and to evaluate how they can (or should) be improved, modified or even discarded in favor of a different approach.

The Unique Characteristics of AI-Based Software Technologies

> Victory or defeat in the air or in space at the human scale is likely to be determined by which combatant has fielded the most advanced AI technology in the areas most crucial to achieving victory. —Frank Kendall, Secretary of the Air Force (Easley, 2025)

Software has rapidly emerged as a transformative force across various sectors, and its significance for modern warfare cannot be overstated.

As noted by Air Force Secretary Frank Kendall in his congressionally mandated *Air Force 2050* Report, "It is likely these areas of advanced military technology will be manifest through the increasingly widespread use of autonomy and automation, in all domains, but especially in space, in cyberspace, and in the air" (Easley, 2025).

For warfighters—the individuals who engage in military operations—AI-based software technologies promise to revolutionize the battlefield by enhancing decision-making, operational efficiency, and survivability. As the global security environment becomes increasingly complex, the integration of software platforms with AI technologies will play a pivotal role in ensuring military readiness and superiority.

A defining characteristic of software-based technologies, including those that contain or exploit Artificial Intelligence (AI) is that they never work perfectly. As sure as the sun rises in the east, software technologies, including AI based software acquisitions, are problematic, generate frustration, and often fail.

That most software applications are riddled with imperfections is not due to deception by software developers. Absent evidence or facts otherwise, software inadequacies are par for the course for an industry that embraces imperfection as a fact of life. Simply put, software systems are rarely deemed complete in the same sense a finished good like a car might be. Instead, software applications are more appropriately considered works in progress or evolutionary drafts that provide increasing increments of better functionality over time.

As the renowned author of *Software Engineering*, Ian Sommerville (2016) stated, the "distinction between [software] development and maintenance is increasingly irrelevant…it is

more realistic to think of software engineering as an evolutionary process where software is continually changed over its lifetime."

Software is imperfect because the cost of perfect software would overwhelm its value and make it economically impossible to purchase. For virtually all software development projects, the industry follows Facebook's Sheryl Sandberg's advice, "Aiming for perfection causes frustration at best and paralysis at worst" (Sommerville, 2016). Yet, the largest single buyer of IT in the world, the Federal Government, will continue to spend approximately $100 billion this fiscal year on IT services and products. Despite the inefficiencies and flaws, software applications usually work well enough, evolve over time to become better, and typically offer a solid return on investment. Imperfections in software have long been accepted as a fact of life. Generally, the inevitable list of flaws slowly, but surely, is corrected over time.

The Government Market for All Types of Software

The government has approximately 2 million software and/or cloud users. For most large commercial IT companies, the government is their single largest consumer. The government has enormous scale. For example, the Veterans Administration, a single agency, currently pays about $600 million per year for just Microsoft products.

The GAO broad estimates are useful but imprecise. Detailed government spending on software and cloud services by brand has never been available. Because most software and cloud are *resold* through government contractors, the brands that are purchased are hidden in the minutia of hundreds of thousands of contracts, which are not available to the public.

Nonetheless, using an average of $17.5 billion per annum, and GAO percentage estimates, the chart below illuminates the approximate market share the U.S. government holds of the top five software or cloud companies (see Table 1 and Figure 1). (Table 1 is an extrapolation by the author from GAO analysis of agency data. The exact percentages are unreported in the literature.)

Table 1. U.S. Government Market Share of Top Five Software or Cloud Companies

Software Company	GAO Gov't Spend Data	Gov't Revenue	Total Corp Revenue 2023	Gov't Revenue
Microsoft	31%	$5,425,000,000.00	$236,000,000,000.00	2%
Adobe	10%	$1,750,000,000.00	$20,000,000,000.00	9%
SalesForce	9%	$1,552,500,000.00	$35,700,000,000.00	4%
Oracle	7%	$1,207,500,000.00	$52,000,000,000.00	2%
ServiceNow	5%	$913,500,000.00	$9,400,000,000.00	10%

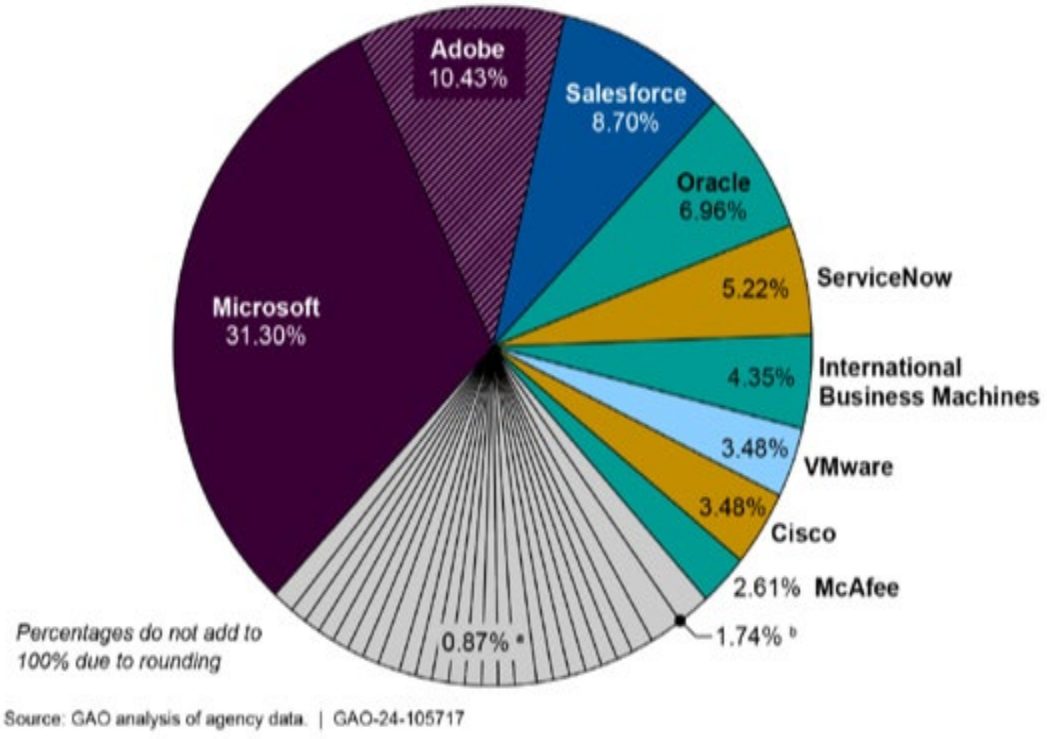

Figure 1. Software Vendors With the Highest Amounts Paid by Agencies for Fiscal Year 2021

In terms of acquisition policy, the government serves in two roles when it acquires pre-existing software-based AI or develops its own custom code. In one role, it procures goods and services to perform its essential functions and fulfill agency missions. During this process, as in the private sector, contracting officers are tasked with negotiating such terms as inspection, acceptance, risk of loss, title transfer, terminations, invoicing, and the like.

In its other role, the federal government "acts as the sovereign"—by driving public policies and imposing unique requirements to advance politically defined societal goals and to further those public policies (Section 809 Panel, n.d., Vol. 3).

These societal policies may indeed promote the public good, but in themselves carry costs that necessarily impose friction in the acquisition process by going well beyond the nature of the product or service being procured.

What are the relevant policy concerns that must be addressed when acquiring AI based software?

First, OMB should issue guidelines to define appropriate government use cases that are appropriate for software technologies that embody AI. For example, it may be a bad idea to give AI control over nuclear launch codes!

Working with tech companies, academia, and other stakeholders, OMB and the agencies can leverage the latest AI advancements and best practices to define appropriate boundaries for appropriate use cases for AI based software contracts. The characteristics of these use cases should reflect a review of such factors as security and ethical use, foster collaboration, provide proper training, and maintain continuous oversight and evaluation.

Second, acquisition tradecraft should adopt a robust evaluation framework to assess AI solutions based on scalability, security, compliance with federal regulations, and transparency.

Third, effective deployment considerations should address up front the best means to integrate AI systems with existing infrastructure, ensuring smooth integration and minimal disruption. New programs must ensure sufficient funding and proper budgeting for AI projects, recognizing their resource-intensive nature.

Fourth, laser-like focus should be applied to the implementation of robust cybersecurity measures. AI systems and sensitive government data must be protected through encryption, access controls, and regular security audits. One approach would be to centralize authority to oversee AI initiatives, ensuring coordination and consistency across federal agencies.

Finally, these contracts must design AI technologies to be fair, unbiased, and respectful of privacy.

Three Procurement Models Currently Predominate for Software Technologies

The current legal disposition of the government's software estate is a hybrid of older contracting models and newer contemporary contracting models stitched together in a complicated patchwork. The power over IT acquisition remains widely diffused. Each agency receives its own budget and makes its own IT acquisition decisions.

As a result, the government's purchase of IT is highly decentralized with little to no government-wide administration. From a technological trends perspective, although the government is not an earlier technology adopter, it has aggressively moved into cloud computing and AI-enabled technologies over the last decade and is now moving into AI embedded software as well.

The mixture of older and newer legal models means that the delivery, licensing, and applicable Terms and Conditions for AI software are as varied as the items purchased. Our research concluded that there are dominant models but no monolithic or one size fits all approach.

Other than the boilerplate terms and conditions that come from shared governmentwide contracts, like the GSA Multiple Award Schedules and NASA SEWP contract vehicles, government purchasers are largely left to their own devices when it comes to AI software acquisitions, including any warranties (or lack thereof).

When the government does review software contract terms, the focus is typically on eliminating potential illegalities in agreements rather than receiving the most useful terms.

Furthermore, because acquisition authorities are agency-based, and agencies only make large software acquisitions every three to five years, they are largely outgunned by the software providers who sell to consumers every day. There is a marked asymmetry of knowledge and expertise, in favor of software companies, despite the enormous spending power of the government. This asymmetry is particularly acute when it comes to AI software acquisitions. It is not unusual for the Contracting Officer responsible for the purchase of AI based software to be more focused on adherence to the procurement rules than on an understanding of the technologies being purchased (Section 809 Panel, n.d., Vol. 3).

At present, there are three dominant acquisition models for AI software: perpetual licensing models, services-based (SaaS) acquisitions based on the cloud, and bespoke (or custom built) AI based software development models. We will discuss the status quo and offer some recommendations regarding each of these prevailing acquisition models.

Existing Contract Models for All Forms of Software: Perpetual License Approach

A common approach to software acquisitions by government is the perpetual license model. Under the perpetual licensing model, the government purchases the right to use the software in perpetuity and then pays an annual maintenance fee to get the latest updates to the code. This is the traditional legal licensing style construct that governed the first generation of software distribution. It reflects significant historical development around "commercial Off the Shelf" or COTS procurements under FAR Part 12.

This "commercial first" acquisition policy is important because of the fashion in which it has been implemented. To meet this policy, the government is charged to adopt standard commercial contract terms, which presumably include industry standard warranty disclaimers and liability cap.

FAR Part 12 is the regulatory and contractual implementation of the statutory mandate (GSA, n.d.).

FAR Part 12 is intended to promote competition, save the government money, and expedite the acquisition of commercial items, including specifically commercial software products. It is also intended to encourage businesses to sell their products and services to the government by providing them with a more streamlined and efficient contracting process.

Under these provisions, the Government is charged to start negotiations with the standard, commercial terms and conditions offered in the private sector by the cybersecurity technology provider. For COTS technologies, the government *must adopt commercial terms and conditions in existing technology agreements unless they violate federal procurement law* (GSA, n.d.).

Commercial software acquisition guidelines are set forth in Federal Acquisition Regulation (FAR) Part 12, which outlines policies and procedures for the acquisition of commercial items by 227.7202-1 Policy.

> "(a) Commercial computer software or commercial computer software documentation *shall be acquired under the licenses customarily provided to the public unless such licenses are inconsistent with Federal procurement law or do not otherwise satisfy user needs*" (FAR 48 C.F.R. 52.227-19, 2007).

Furthermore, in the later provisions of FAR 227.7202-3, the Government is expressly cautioned to not overreach when demanding additional rights and concessions from commercial technology vendors.

The intent of the FAR Part 12 Commercial Items regime is for the government to get the benefit of purchasing ordinary consumer goods and services as closely as possible to the way ordinary consumers purchase the exact same goods and services.

AI issues are relevant under this model, particularly on-premises, perpetually licensed AI based software using perpetual licenses. Here in general, all warranties of any kind are typically disclaimed using the prescription for warranty disclaimers found in Article 2 of the UCC.

Controversially, this type of acquisition has often been equated to the purchase of a tangible good for legal analysis or Uniform Commercial Code (UCC) purposes. Here the user takes possession of the software and may use it in any way that does not violate the usage restrictions of the license agreement. Often this software is used on-premise in facilities controlled by the user. Typical usage restrictions mandate against illegal copying with concern for limiting usage to only those who have been granted permission per the agreement.

SaaS Based Software Acquisitions

As with the technologies themselves, the procedures and techniques used by government to acquire and deploy AI based software technologies have not stood still.

While the basic framework embodied in FAR Part 12 still forms an important structure for these types of software technology procurements, agencies have increasingly adopted and evolved other approaches.

One fundamental shift that is occurring within the industry itself (and therefore the fashion in which government acquires and deploys AI based software technologies), is a fundamental shift from FAR Part 12 commercial product offerings to Software as a Service (SaaS) subscription models.

Since the early 2000s, with the advent of cloud computing, usage and legal models have shifted to a services-based paradigm. The user agrees to pay an annual subscription fee or a periodic consumption-based fee. Here the item being purchased is more readily equated to a service offering. The user never takes possession of the software. It is exclusively controlled by the software provider. The provider maintains it and upgrades it on its own schedule. The software resides outside the four walls of the agency. The user accesses it via a web browser. The user only pays for the actual usage of the application. In the strictest sense (although, not always followed) users pay after consumption of the software service in arrears, on a monthly, or other periodic basis.

As the industry evolves in turns of its business and product delivery model(s), the majority of AI vendors have begun to coalesce around the SaaS model. Users do not control the hosting of the application. Basically, the SaaS model restructures government requirements regarding the manner of their fulfillment. SaaS acquisition models fulfill government requirements by turning them into service offerings under service ordering agreements (called “Service Level Agreements” or SLAs).

Acquiring SaaS creates some unique problems from an acquisition policy perspective. For one thing, these technological offerings are delivered as a service, not as a product. The basic structure of any warranty under these forms of agreement must necessarily differ. The characteristics of service contracts differ substantially from the procurement of tangible goods. Warranties must be adapted to service delivery models instead of tangible product deliveries.

First, unlike product offerings, such issues as warranties, risk of loss, and delivery terms do not neatly translate to service offerings, where performance quality, expertise, and duration are more critical factors.

Second, service contracts often involve considerations like skill, diligence, and outcome, which are subjective and cannot be standardized as easily as terms for products. The performance of services is evaluated based on different criteria, such as professional standards and specific client needs.

Third, services are typically governed not by the UCC but by caselaw applicable to services contracts, which deal with individual fact patterns and therefore provide the necessary flexibility to address the unique aspects of service agreements. Common law allows for a more nuanced and tailored approach, considering factors like implied duties and reasonable expectations.

Challenges for SaaS-Based AI Acquisitions

Given the trend to broader SaaS acquisition models for AI software procurements, two fundamental problems must be addressed.

First, these models are essentially “subscription services.” A subscription model is just what the name entails—the payment of a subscription fee, often up front, for a set period, tied to a number of “seats,” for access and use of the service offering.

The use of subscription models was initially created for research and library services—such as LEXUS/NEXUS or Bloomberg. The model has spread, however, beyond these types of service offerings to encompass subscription-based AI technologies.

In the context of government procurement, subscription models have several structural problems. First off, the government’s archaic and cumbersome appropriations process prevents the government from availing itself of multi-year discounts, which often provide significant cost savings.

To address these concerns, the GSA (2024) recently issued a procurement memo that clarified that a subscription payment was not to be deemed an impermissible “advance payment” banned by the Anti-Deficiency Act. This acquisition memo clarified that upfront payments for software licenses accessed via Software-as-a-Service (SaaS) do not constitute advance payments under specific conditions. This guidance allows federal agencies to use subscription-based pricing models for SaaS offerings without violating advance payment regulations.

The conditions specified in the memo include:

- The software must be accessible immediately upon payment.
- The procurement must be on a fixed-price or fixed-price with an economic price adjustment basis.
- The billing model must not be based on usage or consumption metrics other than quantity. (GSA, 2024).

This clarification aims to align GSA’s procurement policies with modern software delivery methods, making it easier for agencies to adopt cloud-based solutions and streamline the acquisition process.

The second major challenge with SaaS based acquisition models is the warranty problem. Warranties are critical components of all forms of service offerings and provide customers with assurance and security regarding the quality and reliability of the services they purchase.

In the private sector, services warranties take on a distinctly different character. A typical services warranty from a major hyperscaler is as follows:

> 8. Disclaimers.
> THE SERVICES AND AWS CONTENT ARE PROVIDED “AS IS.” EXCEPT TO THE EXTENT PROHIBITED BY LAW, OR TO THE EXTENT ANY STATUTORY RIGHTS APPLY THAT CANNOT BE EXCLUDED, LIMITED OR WAIVED, WE AND OUR AFFILIATES AND LICENSORS (A) MAKE NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE REGARDING THE SERVICES OR AWS CONTENT OR THE THIRD-PARTY CONTENT, AND (B) DISCLAIM ALL WARRANTIES, INCLUDING ANY IMPLIED OR EXPRESS WARRANTIES (I) OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR QUIET ENJOYMENT, (II) ARISING OUT OF ANY COURSE OF DEALING OR USAGE OF TRADE, (III) THAT THE SERVICES OR AWS CONTENT OR THIRD-PARTY CONTENT WILL BE UNINTERRUPTED, ERROR FREE OR FREE OF HARMFUL

COMPONENTS, AND (IV) THAT ANY CONTENT WILL BE SECURE OR NOT OTHERWISE LOST OR ALTERED.
9. Limitations of Liability.
9.1 Liability Disclaimers. EXCEPT FOR PAYMENT OBLIGATIONS UNDER SECTION 7, NEITHER AWS NOR YOU, NOR ANY OF THEIR AFFILIATES OR LICENSORS, WILL HAVE LIABILITY TO THE OTHER UNDER ANY CAUSE OF ACTION OR THEORY OF LIABILITY, EVEN IF A PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LIABILITY, FOR (A) INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES, (B) THE VALUE OF YOUR CONTENT, (C) LOSS OF PROFITS, REVENUES, CUSTOMERS, OPPORTUNITIES, OR GOODWILL, OR (D) UNAVAILABILITY OF THE SERVICES OR AWS CONTENT (THIS DOES NOT LIMIT ANY SERVICE CREDITS UNDER SERVICE LEVEL AGREEMENTS).
9.2 Damages Cap. EXCEPT FOR PAYMENT OBLIGATIONS UNDER SECTION 7, THE AGGREGATE LIABILITY UNDER THIS AGREEMENT OF EITHER AWS OR YOU, AND ANY OF THEIR RESPECTIVE AFFILIATES OR LICENSORS, WILL NOT EXCEED THE AMOUNTS PAID BY YOU TO AWS UNDER THIS AGREEMENT FOR THE SERVICES THAT GAVE RISE TO THE LIABILITY DURING THE 12 MONTHS BEFORE THE LIABILITY AROSE; EXCEPT THAT NOTHING IN THIS SECTION 9 WILL LIMIT YOUR OBLIGATION TO PAY AWS FOR YOUR USE OF THE SERVICES PURSUANT TO SECTION 3, OR ANY OTHER PAYMENT OBLIGATIONS UNDER THIS AGREEMENT. (Amazon Web Services, n.d.; *the language is in all caps by operation of law*)

This hyperscaler disclaims all warranties and sells its services strictly “as is,” meaning they make no claims regarding its application or use. Should a user successfully bring any claim under any legal theory, this hyperscaler limits the amount of damages to 12 months of previous payments.

So far, no major player in the cloud services space has offered anything substantially different.

Other typical SaaS service warranties can be broadly categorized into several types, each serving different purposes and offering varying levels of coverage and assurance:

Performance Warranties

These warranties guarantee that a service will be performed to a specific standard or within a certain timeframe. For example, an IT support service might guarantee a certain level of system uptime or a response time for critical issues.

Satisfaction Guarantees

These warranties promise that customers will be satisfied with the service provided, offering remedies such as refunds or repeat services if the customer is not satisfied. For instance, a cleaning service might offer a satisfaction guarantee, pledging to re-clean the space if the customer is not happy with the initial service.

Service Level Agreements (SLAs)

SLAs are common in industries such as telecommunications and IT services. They outline the expected level of service, including metrics like response times, availability, and performance standards. SLAs are often detailed contracts that specify penalties or compensation if the service provider fails to meet the agreed standards.

Maintenance Warranties

These warranties ensure that regular maintenance services and updates will be provided and to keep equipment functioning optimally. Maintenance warranties often include scheduled inspections, upgrades and repairs.

Other categories of service warranties address the personnel, how the work is to be performed, and establish a contractual floor (they will perform the work "in a good and workmanlike fashion") regarding the conduct of the service.

In reviewing these various types of warranties, SLAs may be the most helpful as they directly address the quality of the services provided. They embody specific, measurable metrics to which the service provider must adhere. If warranty liability limitations were to be applied to SLAs, it has the potential to create "direct line of sight" between the actual performance of the service offering and the fashion in which the underlying code base was designed and implemented.

But clearly some kind of "content warranty" should be reasonable. This form of warranty might provide that the subscription content is up-to-date, i.e., "current, accurate and complete" (the standard for cost and pricing information submitted under cost reimbursement contracts and a well understood concept in government contracting law).

It might also warrant that the content may be lawfully provided to the end use, in the sense that it does not infringe upon, or impermissibly contain, content that is protected by a third parties intellectual rights.

What may be a "bridge too far," however is any kind of warranty that purports to address specific outcomes based upon the use or deployment of the subscription content. Of course, no software producer can warrant the *outcome* achieved by application of a service offering, i.e., the specific results obtained by an agency when it utilizes the service in performance of the agency mission. There are simply too many variables and interdependencies governing specific outcomes. These concerns apply in spades when AI capabilities enter the picture.

No producer can offer a warranty regarding how its service *is applied* to an agency mission set and how it used in practice by a federal customer. Software producers cannot be held responsible for the job performance of federal employees.

An "outcomes-based" warranty would most likely be deemed a stretch, given that the service provider has no control (or even visibility) into the infinite ways that the subscription content may be put to use by the government or any other end user.

DoD: A Closer Look at Software Acquisition Policy for the Warfighter

Within the DoD, these existing procurement models have played out in the context of a dismal government-wide history of major (and very expensive) IT project failures. In 2015, the Standish Group analyzed over 25,000 software application development projects across both industry and the government and determined that 24% of all custom software development projects for government failed, while 55% were "challenged" and only 21% were considered successful. Only 40% of all projects, government or otherwise, were completed on time. Only 44% were completed on budget (Standish Group, 2015).

That only 40–44% of development projects finished on time and on budget is important. Time is always the invisible force behind every project. It is not unusual for software development projects to be overloaded with features, ultimately making it impossible to meet optimistic deadlines. Every development project has a notional schedule, and as difficulties are encountered and projects begin to fall behind, sometimes there is external pressure to meet the

deadline, rather than 100% of the requirements. According to the Standish Report, this decision process would likely occur in about 60% of bespoke custom projects (Standish Group, 2015).

As described in *The Washington Post*, in 2015, the Government Accountability Office (GAO) listed a litany of failed federal and DoD IT projects, including:

- DoD's Expeditionary Combat Support System: canceled after failure to deploy and more than $1B expended.
- DHS's Secure Border Initiative Network Program: canceled after $1B obligated, could not meet viability standards.
- VA's Financial and Logistics Integrated Technology Enterprise Program: terminated after $609M expended.
- NOAA's National Polar-orbiting Operational Environmental Satellite Systems: terminated after approximately $5B spent. (Ravindranath, 2014).

None of these canceled projects were implicated in fraud, were the subject of a qui tam (whistleblower) proceeding, or were mired in criminal controversy. They failed for the reason most fail: because they are hard, because it is software and because humans are fallible. To stay ahead, the DoD recognized that it must build strong, secure software quickly. This means software that is reliable, safe from cyber threats, and delivered fast enough to maintain its competitive edge.

Recognizing this lengthy track record of IT development failure, in 2022, the DoD pushed out a comprehensive DoD Software Modernization Policy (DoD, 2022). This policy was necessitated by the very poor track record by the government in developing its own government-unique IT systems. As previously described, it is a record unsullied by success.

The DoD's software modernization policy established several key metrics to advance modern agile software deployment across the DoD enterprise (DoD, 2022).

Security, Stability, and Speed Matter Equally

Speed is important, but the DoD can't sacrifice security or reliability. DoD software must be dependable and resistant to cyber threats while still being built efficiently. The key is adopting modern development methods that focus on security and performance at every step.

Smarter Use of Cloud and Data

Cloud technology and data management are essential to software modernization. The DoD must move quickly to the cloud and follow best practices for handling data to improve capabilities and decision-making.

Enterprise-Wide Solutions

The DoD acknowledged that it has to be smart about spending. Instead of funding multiple overlapping software projects, the focus should be on shared enterprise solutions that help the entire department save money while still being effective.

No One Left Behind

Software modernization isn't just about new technology, it's about people. DoD leadership must invest in training and upskilling the workforce so that employees can take full advantage of new tools and automation.

More Than Just Code

Writing code is only one part of software modernization. The DoD understood that it must also streamline policies, contracts, and processes to make it easier to develop, buy and implement software efficiently.

The DoD also recognized that no matter how software is acquired, it must be continuously updated. Software isn't a one-time purchase—it requires constant improvements to stay secure and effective.

Use Design Patterns

These are pre-made solutions that speed up development and ensure security. Automating common software tasks—like setting up cloud environments—makes software delivery faster and more consistent.

Improve Cloud Contracts

The DoD must have easy, fast access to commercial cloud services. This means improving the way cloud contracts are structured to avoid delays that could put military operations at risk.

Make Software Purchases More Flexible

The DoD needs faster, more flexible ways to buy and fund software projects, ensuring that critical technology is delivered without unnecessary delays.

DoD Implements Its Modernization Policy by Adopting Agile Development Techniques

Implementing the new software modernization policy is a sea change by the DoD in terms of its core software development techniques. When it comes to being a special case for software, the DoD is often unique. Putting aside embedded code for weapons systems, much of the DoD software estate is fundamentally different from the commercial market. For many years, the DoD pressed forward with custom built software using the so-called "waterfall" design methodology (DoD, 2020).

"[A traditional waterfall development lifecycle builds the entire product with a single delivery at the end, thereby increasing the risk of either delivering the wrong product or not successfully delivering a product. End-user feedback is generally not received until the full solution is developed, and the end-user receives no value potentially for years" (DoD, 2020).

As a result of these numerous IT program failures, the government concluded that software should no longer use a "waterfall" design approach but should be developed and delivered in iterative stages under a process called agile or rapid prototyping. This process calls for the development of software in "bite sized" portions under a protype type of contracting structure.

Agile software development is an iterative and flexible approach that focuses on collaboration, adaptability, and continuous improvement. Unlike traditional waterfall models, which rely on extensive upfront planning and rigid processes, Agile allows teams to respond quickly to changing requirements and customer needs. Agile methodologies, such as Scrum, Kanban, and Extreme Programming (XP), emphasize incremental development, where software is built in small, manageable iterations known as sprints. This approach ensures that teams can make adjustments throughout the development cycle, leading to more efficient and high-quality software solutions.

The purpose is for the product to be developed and tested iteratively so that it may "fail early/fail fast" if it is inadequate or improperly designed. The focus is on rapid failure and revisions.

Agile methods outperform traditional software development approaches for several key reasons (DoD, 2020).

Flexibility and Adaptability

Agile's iterative nature enables teams to reassess priorities and make changes as needed. This ensures that software remains relevant and aligned with evolving business needs and technological advancements.

Enhanced Collaboration and Communication

Agile fosters a culture of teamwork and transparency. Regular meetings, such as daily stand-ups and sprint reviews, keep all stakeholders informed and engaged, reducing misunderstandings and promoting shared goals.

Faster Time-to-Market

By breaking projects into smaller increments, Agile teams can develop, test, and deploy functional software quickly. This allows organizations to introduce new features and improvements faster, maintaining a competitive edge.

Improved Product Quality

Continuous testing and feedback integration ensure that defects are identified and addressed early. Techniques like automated testing, pair programming, and frequent code reviews enhance software reliability and performance.

Higher Customer Satisfaction

Agile prioritizes customer involvement, ensuring that their feedback is incorporated throughout the development process. This leads to a product that better meets user needs and expectations, fostering trust and long-term client relationships.

The DoD implemented an agile design approach by creating six new acquisition pathways, including specifically a software acquisition pathway to accelerate the development and delivery of software (DoD, 2020).

In 2020, the DoD issued a detailed guide regarding its current thinking about how best to contract for agile software development using the software acquisition pathway (DoD, 2020). The Guide sought to record "lessons learned" and best practices from the Congressionally mandated Agile pilot program codified in the FY2028 National Defense Authorization Act. It emphasized that Agile development requires a shift from traditional waterfall contracting to more flexible, modular approaches that align with Agile principles of emergent and adaptive planning.

> One of the key differences between an Agile project and traditional waterfall project is that Agile principles align with emergent and adaptive planning and design principles, whereas waterfall projects are focused on predictive planning and upfront design readiness. Agile teams expect to learn through continuous experimentation and continuous delivery. The Agile teams anticipate the need for change or modification of requirements and design as they learn from the fast feedback loops. (DoD, 2020)

Although noting that "there is no single recommended contracting strategy for an Agile software development effort," the Guide advocates for modular service contracting principles, noting that modular service contracts are preferred in Agile environments, allowing the government to

acquire contractor expertise and skillsets while enabling continuous reprioritization without contract modifications (DoD, 2020).

> For these reasons, the acquisition of services over specific product deliverables becomes an important consideration in Agile development efforts. Traditional IT acquisition programs contract to deliver a product capability based on a defined set of "complete" requirements. (DoD, 2020)

The Guide embraces modular services contracting as a means of reducing risk by managing smaller contracts, incentivizing performance, and enabling continuous innovation.

Modular services contracting aims to develop discrete capabilities fitting into an overall technology vision based upon smaller acquisition increments. These increments should be simpler to manage than a single "big bang" waterfall approach, allow complex IT objectives to be addressed incrementally to maximize the likelihood of a workable system, and provide for the delivery of testable working solutions, "each of which comprises a system or solution that does not depend on any subsequent increment in order to perform its principal functions."

In terms of acquisition tradecraft, modular services contracts have different attributes and considerations. A modular services contract is based on the performance of contractor labor hours, not delivery of some unitary defined product. The Guide notes that FAR Part 16.5 procedures allow for task orders against existing contracts, such as GWACs, MACs, and agency-specific IDIQ contracts, to acquire platform and integration subscriptions, microservices, and other services.

The Guide recommends shorter periods of performance (one year or less) with options for additional support to avoid long-term commitments if contractor performance is unsatisfactory.

It also discussed intellectual property strategies, seeking to ensure the government only pays for necessary IP, considers deferred ordering, and includes clear guidance on data rights and access in case of early contract termination.

In this context, no form of product warranty can be assigned because the "product" is a hybrid base of code under rapid evolution and change. It would seem in this context that the warranty framework should reflect the fashion and means by which the developers conduct themselves—in other words, a professional services style of warranty.

Progress toward implementing this new software acquisition pathway has been slow, however, with (as of 2024) only 50 programs utilizing the new approach (Obis, 2024).

Recent press reports suggest that the DoD is seeking to optimize the software acquisition pathway to specifically address AI based software developments. All of this changed on March 6, 2025, when the DoD issued a much-anticipated acquisition memorandum squarely addressing a new path forward for DoD-centric software procurements.

DoD Fully Embraces Agile Development Techniques With a New Procurement Concept: OTA Based CSOs

As noted, the DoD has long recognized the need to evolve its software acquisition strategies to keep pace with the rapid technological advancements and increasing demands for agile and efficient military capabilities. In the transcript dated March 7, 2025, titled "Directing Modern Software Acquisition," significant insights were provided regarding the newly issued March 7, 2025, DoD Memorandum entitled *Directing Modern Software Acquisition to Maximize Lethality* (DoD, 2025).

The DoD's new approach leverages the lessons learned from its previous pilot programs as well as the operational history of the Defense Innovation Unit (DIU). It seeks to address deficiencies that arise from using traditional acquisition methods, often leading to delays and poor outcomes. This pathway focuses on rapid deployment and iterative development. By fostering agile-like collaboration between government entities and private sector partners, the DoD aims to accelerate the delivery of mission-critical software.

Recognizing the vital role that the private sector plays in software development, the new pathway emphasizes enhanced collaboration with industry partners. The DoD emphasizes that it is actively seeking to establish partnerships with innovative tech companies, startups, and research institutions to tap into their expertise and cutting-edge technologies. This collaborative approach not only helps to access a broader range of solutions but also encourages knowledge sharing and innovation. As noted in DoD press comments about the new acquisition memorandum:

> So right now, the way the Pentagon buys software is slow, outdated and filled with bureaucracy. Meanwhile, our adversaries are moving fast. This memo is the beginning to fix that, cutting red tape, working more with private industry, getting cutting edge software into the hands of our warfighters quickly before the enemy can adapt. And one of the biggest changes is using flexible contracting tools, CSOs and OTs to speed up innovation and acquisition. (DoD, 2025)

One of the most notable features of the new software acquisition pathway is its full-throated embrace of agile development methodologies. This approach allows for continuous integration and delivery, enabling teams to respond swiftly to changing requirements and user feedback. The framework encourages iterative development cycles, where software can be tested, evaluated, and refined in shorter timeframes. This is crucial in a military context, where the ability to adapt to new threats and operational environments can determine, often, mission success.

Another important aspect of the new pathway is the establishment of cross-functional teams. These teams bring together experts from various domains—including software engineering, cybersecurity, user experience, and operational planning—to collaborate on software development projects. By breaking down traditional silos, the DoD aims to leverage diverse perspectives and expertise, leading to more holistic and effective software solutions.

The new pathway also seeks to simplify acquisition processes by reducing bureaucratic hurdles. This includes streamlining documentation requirements and minimizing the number of approvals needed at various stages of development. By doing so, the DoD aims to empower program managers to make decisions more rapidly, allowing for quicker responses to emerging needs and opportunities.

Furthermore, the development of the new software acquisition pathway promotes the use of open-source software and modular architectures, which can significantly enhance interoperability and reduce costs. By adopting these strategies, the DoD can foster a more collaborative development environment and leverage existing technologies, thus accelerating the timeline for new capabilities.

Specific Contract Structures That Best Leverage the Agile DoD Approach

The DoD software acquisition memorandum establishes OTA-based contracts using Commercial Solutions Opening (CSO) acquisition procedures. As the DoD briefed the press when releasing its software acquisition memorandum:

> So, when we take that software pathway mechanism and we combine it with innovation that DIU has been working in commercial solutions openings, or CSOs, and other transaction authorities, OTAs, we get to the point where now we can expose the program, the software programs, to nontraditional and commercial software developers while we simultaneously lower the barrier for those nontraditional and commercial software developers to get in to defense programs of record. (DoD, 2025)

By way of a quick background, Other Transaction Authority (OTA) based contracts have emerged as a flexible and innovative procurement method, particularly within the DoD and other federal agencies (Defense Acquisition University, n.d.).

These contracts differ significantly from traditional procurement mechanisms by allowing agencies to bypass many of the regulatory constraints associated with the Federal Acquisition Regulation (FAR). OTAs enable rapid prototyping, development, and deployment of cutting-edge technologies by fostering collaboration between government entities, private industry, and non-traditional contractors.

Some of their primary benefits are:

Flexibility and Reduced Bureaucracy

Unlike FAR-based contracts, OTAs are not bound by extensive government regulations, allowing for streamlined negotiation and execution. This flexibility accelerates project timelines and reduces administrative burdens.

Encouragement of Innovation

OTAs are designed to attract non-traditional contractors and startups that may not typically engage with government contracts due to complex regulatory requirements. This encourages fresh ideas and technological advancements.

Rapid Prototyping and Development

One of the primary purposes of OTAs is to support fast-paced research and development (R&D). They enable iterative testing and refinement of prototypes before full-scale production, reducing risks and improving final outcomes.

Collaborative Approach

OTA agreements often involve a high degree of collaboration between government agencies, private companies, and academic institutions. This partnership-driven model fosters a cooperative environment for technological breakthroughs.

Customized Agreement Terms

Unlike standard contracts, OTAs allow negotiators to tailor terms to fit the specific needs of a project, ensuring a more effective alignment between government requirements and industry capabilities.

Also, the Commercial Solutions Opening (CSO) approach is a procurement strategy used primarily by government agencies, such as the DoD, to acquire innovative commercial products, services, or technologies in a streamlined and flexible manner. It's designed to encourage competition, attract non-traditional vendors (e.g., startups or companies that don't typically work with government), and expedite the acquisition process compared to traditional methods like the Federal Acquisition Regulation (FAR) based solicitations. The CSO process emphasizes outcome-based requirements and leverages commercial market practices rather than rigid government specifications (Defense Acquisition University, n.d.).

A CSO approach encompasses the following attributes:

1. Broad Solicitation

Instead of a detailed, prescriptive request for proposals (RFP), a CSO issues a broad problem statement or desired outcome (e.g., “enhance cybersecurity for cloud-based systems”). Vendors are invited to propose innovative commercial solutions that address the need.

2. Flexible Evaluation

Proposals are evaluated based on their technical merit, feasibility, innovation, and cost-effectiveness rather than strict compliance with predefined specs. Peer reviews or expert panels often assess submissions.

3. Commercial Item Focus

Solutions must qualify as “commercial items” under FAR Part 12, meaning they’re already available in the marketplace, customized from commercial products, or developed using commercial practices.

4. Simplified Process

The CSO skips some of the bureaucratic steps of traditional procurement, allowing faster award timelines. Contracts can be fixed-price, other transaction agreements (OTAs), or similar mechanisms.

5. Iterative Engagement

Agencies can down-select vendors through phases (e.g., concept papers, pitches, prototypes), fostering collaboration and refinement before final awards.

When applied directly to AI embedded software procurements, the CSO approach aligns well with the fast-evolving nature of AI software development and the commercial AI software market. Here’s how it works in practice:

1. Problem-Driven Solicitation

Instead of specifying a particular software stack or architecture (e.g., “must use Java and run on Oracle databases”), the agency might issue a CSO stating, “We need a scalable solution to manage real-time data analytics for 10,000 users.” This invites vendors to propose diverse solutions, such as SaaS platforms, custom-built tools, or open-source adaptations.

2. Encouraging Innovation

Software startups or companies with cutting-edge AI, cloud, or DevSecOps tools—often excluded from traditional procurements due to complex compliance requirements—can pitch their products. For example, a vendor might offer a machine learning-based cybersecurity tool already used by private industry, adapted for government needs.

3. Evaluation Flexibility

Proposals might include demos, proof-of-concept code, or access to existing platforms rather than lengthy documentation. Evaluators could assess based on usability, scalability, security, and alignment with commercial best practices, rather than checking boxes for government-specific standards.

4. Commercial Software Focus

The CSO prioritizes off-the-shelf (COTS) or minimally customized software. For instance, an agency might procure a commercial project management tool like Jira or a cloud platform like AWS, tailored slightly for specific security protocols. Open-source software could also qualify if it’s commercially supported or widely adopted.

5. Rapid Acquisition

Traditional software procurements can take months or years due to detailed RFPs and protests. A CSO might award a contract in weeks by selecting a vendor after a pitch day or prototype phase. Example: An agency needs a collaboration tool. After a CSO, vendors like Slack, Microsoft (Teams), and a niche competitor submit proposals. The agency picks one after a quick demo round.

6. Phased Approach

For complex software needs (e.g., a new logistics system), the CSO could start with white papers, move to a prototype phase, and end with a full deployment contract, reducing risk and allowing iterative feedback.

Applying Best Practices, Specific Recommendations

Negotiating an Agile-based software development contract requires careful consideration to ensure that both parties align on expectations, responsibilities, and outcomes. Unlike traditional fixed-price contracts, Agile contracts must accommodate evolving requirements and continuous feedback loops. Here are some of the best practices utilized in the commercial sector to draft agile development contracts:

1. Define Project Scope with Flexibility

While Agile development emphasizes adaptability, the contract should establish an initial project vision and objectives. Instead of rigidly defining every feature upfront, the contract should outline high-level deliverables and expected business outcomes while allowing room for evolving requirements. This approach ensures that both parties share a common understanding of the project's goals without stifling Agile's iterative nature.

2. Establish a Transparent Pricing Structure

A well-structured Agile contract should balance financial predictability with the flexibility required for iterative development. Common pricing models include time-and-materials (T&M), capped T&M, and fixed-price per sprint. Clearly, defining how costs will be measured and adjusted throughout the development process helps prevent financial disputes while ensuring the development team is adequately compensated for their work.

3. Incorporate Change Management Mechanisms

Change is a fundamental aspect of Agile development, and the contract should reflect this reality. It is essential to include mechanisms for managing changes in scope, priority, or functionality. Agreements should specify how changes will be documented, evaluated, and approved, ensuring that both parties remain aligned throughout the project lifecycle.

4. Define Roles and Responsibilities Clearly

Effective Agile contracts clearly define the roles and responsibilities of all stakeholders, including the development team, product owner, and client representatives. This prevents misunderstandings and ensures that collaboration remains productive. The contract should also outline expectations regarding participation in Agile ceremonies, such as sprint planning, daily stand-ups, and retrospectives.

5. Align Incentives with Performance and Outcomes

To foster a successful partnership, the contract should include incentive mechanisms that align with project success. Performance-based payments, milestone achievements, and bonuses for early or high-quality delivery encourage both parties to focus on delivering value rather than merely completing predefined tasks.

6. Ensure Clear Communication and Dispute Resolution

Given the iterative nature of Agile projects, maintaining open and transparent communication is critical. The contract should establish regular check-ins, feedback loops, and escalation procedures to resolve disputes efficiently. Well-defined conflict resolution mechanisms can help mitigate risks and ensure a smoother development process.

Negotiating an Agile software development contract requires a balance between flexibility and structure. By defining a high-level scope, establishing a transparent pricing model, incorporating change management processes, and ensuring clear communication, both parties can maximize the benefits of Agile development. Well-crafted contracts facilitate collaboration, enhance adaptability, and improve the chances of project success, making them an essential component of modern software development agreements.

When implementing its new CSO/OTA acquisition methodology, the DoD would be well served to adopt the best practices pioneered by industry in structuring its software development agreements, especially those that embed or incorporate AI-based technologies.

Conclusion

The introduction of the new DoD software acquisition pathway marks a significant shift in how the Department approaches the development and procurement of software solutions.

By embracing Agile methodologies, fostering collaboration, simplifying processes, and engaging with industry, the DoD aims to create a more dynamic and responsive acquisition environment.

This transformation is essential for ensuring that the U.S. military remains at the forefront of technological advancements, capable of addressing the evolving challenges of modern warfare and delivering national security.

As these changes are implemented, the hope is that they will lead to more effective software solutions that enhance the operational capabilities of the armed forces, ultimately contributing to mission success in an increasingly complex global landscape.

References

Amazon Web Services. (n.d.). *Standard warranty disclaimer*. https://aws.amazon.com/legal/

Defense Acquisition University. (n.d.). *CSO vs prototype OT vs procurement for experiments*. Retrieved March 27, 2025, from https://aaf.dau.edu/aaf/contracting-cone/cso-v-otas-v-2373/

DoD. (2020, February 27). *Agile software acquisition guidebook: Sustainment strategy*. Defense Acquisition University. https://www.dau.edu/sites/default/files/Migrated/CopDocuments/AgilePilotsGuidebook%20V1.0%2027Feb20.pdf

DoD. (2022, February 3). *Department of Defense software modernization strategy*. https://media.defense.gov/2022/Feb/03/2002932833/-1/-1/1/DEPARTMENT-OF-DEFENSE-SOFTWARE-MODERNIZATION-STRATEGY.PDF

DoD. (n.d.). *DFARS 212.102(a)(iii)*. Defense Acquisition University. https://aaf.dau.edu/aaf/contracting-cone/commercial-items/

DoD. (2025, March 7). '*Directing modern software acquisition to maximize lethality' memo background briefing*.

https://www.defense.gov/News/Transcripts/Transcript/Article/4113458/directing-modern-software-acquisition-to-maximize-lethality-memo-background-bri/

Easley, M. (2025, January 13). SECAF Kendall, looking out to 2050, predicts war winners will be combatants with the best AI. *Defense Scoop*. https://defensescoop.com/2025/01/13/frank-kendall-air-force-2050-predicts-war-winners-will-be-side-with-best-ai/

GSA. (n.d.). *Federal Acquisition Regulation (FAR) part 12 - acquisition of commercial products and commercial services*. Acquisition.gov. Retrieved March 27, 2025, from https://www.acquisition.gov/far/part-12.

FAR, 48 C.F.R. 52.227-19 (2007). https://www.acquisition.gov/far/52.227-19

Federal Acquisition Streamlining Act of 1994, Pub. L. No. 103–355, 108 Stat. 3243 (1994). https://www.govinfo.gov/content/pkg/COMPS-1008/pdf/COMPS-1008.pdf

GAO. (2024). *GAO analysis of agency data* (GAO-24-105717). https://www.gao.gov/products/gao-24-105717

GSA. (2024, March 15). *GSA acquisition memo: Subscription-based pricing* (Acquisition Letter MV-2024-01). https://www.gsa.gov/system/files/MV-2024-01.pdf

Obis, A. (2024, October 10). DoD considers faster acquisition pathway for AI. *Federal News Network*. https://federalnewsnetwork.com/defense-main/2024/10/dod-considers-faster-acquisition-pathway-for-ai/

Ravindranath, M. (2014, June 15). GAO: $10 billion worth of current federal IT contracts could fail. *The Washington Post*. https://www.washingtonpost.com/business/on-it/gao-10-billion-worth-of-current-federal-it-contracts-could-fail/2014/06/13/1e6ee728-f24f-11e3-9ebc-2ee6f81ed217_story.html

Section 809 Panel. (n.d.). *Section 809 panel*. Defense Technical Information Center. https://discover.dtic.mil/section-809-panel/

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson. https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf

Standish Group. (2015). *CHAOS report*. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net