leidos

Enabling Continuous Modernization for Army Software Systems

EXECUTIVE SUMMARY

"The primary end state of the 2021 update to the Army Modernization Strategy (AMS), nested with the 2018 Army Strategy, is a modernized Army ready to conduct Multi-Domain Operations (MDO) as part of an integrated Joint Force. The MDO concept describes how the Army will support the Joint Force in the rapid and continuous integration of all domains of warfare – land, sea, air, space, and cyberspace – to deter and prevail as we compete short of conflict, and fight and win if deterrence fails." – 2021 ARMY MODERNIZATION STRATEGY

Continuous Modernization is an Army-wide endeavor, requiring Army-wide coordination, to continually provide rapid delivery of enhanced capabilities to soldiers. The Army recognizes that industrial age, serial processes cannot keep up with ever-changing missions, operating systems, hardware, communications protocols, networks, message formats, cyber threats, vulnerabilities, data sources and formats.[1] Therefore, the Army and the Joint Force must continue the shift from Industrial to Information Age approaches to modernization to accelerate the acquisition of mission software and systems.[2]

The Army understands the need to accommodate continuous new requirements as the norm, not the exception, based on evolving mission needs, emerging threats, soldier feedback, and usage analysis. This requires a continuous, parallel, incremental cycle of discrete, manageable, mutable, and containerized outputs to effectively meet new requirements at the point of need and the speed of relevance.

In this paper we describe a soldier-centric, subscription-based business model using the existing acquisition processes. This approach harnesses the power of the Army's cloud computing environments while providing a modernization on-ramp for fielded legacy applications. Further, continuous delivery coupled with a robust Zero Trust security posture will provide the flexibility, scalability, security, and oversight required to field, monitor, adapt, and enhance mission software and systems at the speed of relevance.

To accomplish these objectives, we recommend the Army:

- Exploit the Army Cloud Ecosystem
- Incrementally Transition Legacy Applications to Cloud Native Services
- Employ Continuous Delivery Across All Phases of the Capability Lifecycle
- Embed Zero Trust Security Throughout the Lifecycle
- Adopt a Subscription-Based Business Model
- Align Contracts to Continuous Modernization

These actions will enable a continuous stream of enhancements, fixes, and adaptations to develop, deploy, and maintain software in today's hyper-velocity environment. The recommended approaches will support the Army's goal to "…use adaptive acquisition approaches that leverage the full scope of Congressional authorities, such as Middle Tier Acquisition, to accelerate development, production, and delivery of materiel capabilities" and "use innovative contracting tools, such as Other Transaction Authorities and Cooperative Research and Development Agreements, to foster innovation..." [3]

In addition, the recommendations support the Honorable Doug Bush's, Assistant Secretary of the Army for Acquisition, Logistics and Technology priorities to focus on rapid delivery, accelerate software acquisition, instantiate cybersecurity and supply chain security, and enable realistic and early testing of systems. [4]

There are no technical barriers to achieving this -- rather, existing processes and procedures can be modified to rapidly instantiate a robust, agile, and rapid critical capability development and deployment pipeline.

1. Exploit the Army Cloud Ecosystem

Cloud computing enables convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Edge to Cloud (E2C) adds resilient commodity edge hardware, software, and communications to securely connect sensors, mission software, services, and data via a vendor-agnostic infrastructure in edge nodes that can operated disconnected in disrupted, disconnected, intermittent and low-bandwidth (DDIL) environments. E2C edge nodes enable connectivity to existing systems across heterogeneous and disconnected environments.

The *2020 The Army Cloud Plan* stated: "The Army cannot maximize its modernization strategy without the cloud, which is the backbone for artificial intelligence." [5] This requires a cloud ecosystem that spans the enterprise down to low SWAP-C nodes that extend the reach of cloud computing power, storage, and analysis. The Leidos Edge to Cloud (E2C) ecosystem enables centralized deployment and maintenance using the same software, analytics, datasets, communications, security, information assurance, policies, procedures, interfaces, and displays from the enterprise cloud to the tactical edge. A distributed cloud built on nodes will permit rapid deployment of fixes, enhancements, and adaptations produced through the Continuous Modernization model.

Austere warfighting environments, denied, degraded, intermittent or limited-bandwidth (D-DIL) communications, security concerns, distributed devices, and severe size, weight, power, and size, weight, power, and cost (SWaP-C) constraints have limited the extension of cloud computing to the tactical edge where the fight occurs. To address this gap Leidos partnered with several leading commercial vendors in the cloud computing space and built an ecosystem upon which data, mission software, and AI/ML analytics can be deployed, managed, shared, and sustained on edge nodes. The Edge to Cloud (E2C) construct is designed to be extensible to hybrid cloud networks based on AWS, Microsoft Azure, Google Cloud Platform, and other cloud service providers. E2C offers significant advances in processes and technologies that can immediately augment any organization's current and future missions. The E2C vendor-agnostic cloud-enabled ecosystem enables continuous agile development, provisioning, and sustainment (DevSecOps) to augment existing TTPs, SOPs, and legacy platforms and applications. This approach will:

- Engage warfighters in continuous, rapid, iterative mission capability enhancement
- Enable integration of best of breed technologies, processes, interfaces, and datasets
- Avoid platform-centric approaches to data flows and processes
- Eliminate vendor lock
- Encourage continuous improvement
- Leverage existing TTP, SOP, and platforms
- Reduce risk, cost, and schedule to provide enhanced capabilities

2. Transition from Legacy Applications to Cloud Native Services

There are graded levels of application deployment to the cloud, from using the cloud as a virtual server to containerized components to cloud-native microservices. These levels of maturity permit gradual refactoring of legacy applications over time while harnessing the computer, storage, and communications power of the cloud today.

The simplest and fastest is lift and shift, where the application is deployed in a Virtual Machine (VM) that replicates the application's native environment to include the complete operating system (O/S). A VM provides complete isolation from the cloud host operating system and other VMs while immediately placing the legacy application in the cloud. Though there are some benefits to VM, monolithic applications deployed in this way impose a very large footprint within the cloud host, due to the duplication of operating system (O/S) for each application, the need for modifications to the O/S to run within a specific cloud environment, and the inability to discretely patch, enhance, or change legacy code without affecting the legacy baseline. Therefore, lift and shift is the first step in migrating from legacy to far mor efficient and maintainable services deployed within containers. Containerization breaks up the legacy application into components that only carry the code necessary to run the component (unlike a VM, which typically uses the entire legacy O/S). This approach helps reduce the application footprint, but still depends on smaller monolith code.

Cloud-native code uses Microservices, which consist of multiple loosely coupled components supported by an infrastructure for providing application services. Microservices provide a cloud-native integration framework across legacy, hybrid, and cloud-native target-data systems to enhance mission data processing with the long-term objective of refactoring, which replaces legacy code with native microservice-based code incrementally, over time, reducing impacts to training and readiness.

We have categorized microservices into six categories:

- **Connect:** Establishes ability to publish, subscribe, and query via application interfaces.
- **Translate:** Converts from one data format to another.
- **Extend:** Provides new functionality to existing POR.
- **Process:** Applies automated actions to refine, edit, refactor data.
- **Display:** Provides user interface to view data.
- **Monitor:** Records data for subsequent analysis.

The "strangler fig pattern" uses microservices to migrate legacy applications by gradually replacing specific components with services.[6] Refactoring a complex system can be a huge undertaking, whereas gradual migration over time will enhance adoption while minimizing impacts on readiness and operations. This can be aided by maintaining the legacy application façade that intercepts requests and routes these requests either to the legacy application or the new services. Existing features can be migrated to the new system gradually, and soldiers can continue using the same interface. This pattern helps to minimize risk and spread the development effort over time. We recommend employing the cloud maturity model to deploy applications to the cloud and gradually refactor code to cloud-native:

- Deploy legacy application to cloud on virtual machine (VM).
- Provision and manage cloud nodes via Army CI/CD pipeline.
- Develop microservices to connect, extend, and gradually replace legacy functionality.
- Monitor usage to identify gaps, operator feedback, and maintain continuous ATO.
- Extend microservices via continuous enhance, fix, adapt cycles.

This approach creates incremental value immediately, with successive builds delivering more capable, resilient, and secure code with each release. The incremental deployment of microservices also enables rapid shifts in focus to accommodate immediate mission needs, permitting far greater agility than legacy development approaches.

3. Employ Continuous Delivery Across All Phases of the Capability Lifecycle

Continuous Modernization assumes that software is never finished. While the scope may vary, the need for constant updates due to new requirements, uncovered vulnerabilities, new threats, new technologies, and new platforms will be constant. Continuous modernization requires a shift from product delivery to capability provisioning aligned with constant evolution. While the scale of the effort will likely change over time, the need for constant updates due to new requirements, uncovered vulnerabilities, new threats, new technologies, and new platforms will be constant. Further, while there have been numerous attempts to adjust expectations and enforce rigidity into critical software projects, all systems require continuous enhancement and extension (software is never done, it simply meets a threshold of expected capability), and yet existing applications are not readily adaptable to emerging mission demands.

We define continuous modernization as more than episodic releases of new (e.g. "modern") technologies. Rather, it is guided evolution, which is comprised of both reaction to environmental challenges and incremental enhancement of existing capabilities. Therefore, in this paper we define Continuous Modernization as the alignment of an organization's acquisition processes, prioritization adjudication, and technical practices to reduce the gaps between stakeholder expectations and constraints, development throughput, security considerations, and operational user needs through deliberate evolution over time. Continuous Modernization instantiates processes to encourage cross-domain collaboration and introduces a continuous pipeline to produce useful, secure, and adaptable capabilities. This approach encourages experimentation that results in incremental and breakthrough improvements, thus removing the divide between research and development, production, and sustainment.

The commercial software industry has adopted a continuous delivery approach, abandoning monolithic version releases. Today software on PCs and mobile devices is purchased as a subscription, with the "version" under continuous revision.[7] Microsoft's 365 Business Suite is now offered as an annual subscription ranging from three applications (Word, Excel, and PowerPoint) to over a

dozen.[8] Mobile phone users subscribe to apps that are under continuous update release (with settings permitting apps to be updated in the background, unknown to the user). Commercial industry has recognized that constant enhancement, correction, threat mitigation, and adaptation is the rule, not the exception.

In traditional acquisition models a mission need is defined, funds are allocated, a project is initiated, requirements are elicited, and a product produced that meets some percentage of expressed requirements.[9] New requirements are treated as exceptions, and often cause projects to slowly expand due to "scope creep," or "high priority" enhancements are included, and lower priority known vulnerabilities and capability gaps are deferred. At some level of maturity (often defined by funding constraints, not capability thresholds) the system is placed into sustainment where a different group of technicians assumes responsibility for change management and continued development.[10]

Thus, Continuous Modernization is not merely technical (although Continuous Modernization leverages technology) but an extension of the organization of people, processes, and technologies to address ever-changing mission needs, vulnerabilities and threats, and enhancement requirements both rapidly and securely. The approach recognizes the reality of continuous software evolution to accommodate new mission requirements, new threats and vulnerabilities, and new connections within and adaptations to the host environment.[11]

This changes the software acquisition and development process from a linear progression towards arbitrary "completion" to a continuous pipeline of small, manageable, mutable, and containerized products to meet capability needs.

The benefits of this approach only obtain when several key project disciplines are in place. Effective governance implies constant communication, acknowledged stakeholders with defined roles (approve, review, consult, informed), an understanding of requirements, minimal but transparent reporting, and risk management within a disciplined framework throughout the project lifecycle.

4. Embed Zero Trust Security Throughout the Lifecycle

Zero Trust (ZT) was mandated by presidential Executive Order in May 2021 and is a dramatic shift from common practice of allowing an authenticated user unrestricted access to network resources once inside the perimeter. ZT eliminates the notion of trust to protect networks, applications, and data. This contrasts with the traditional perimeter security model, which presumes that bad actors are always outside the network. With ZT all users are presumed to be untrustworthy. ZT continuously verifies every transaction, asserts least privilege, and relies on intelligence, advanced detection, and real-time response to threats. ZT continually reduces the breath, depth, and duration of trust granted to each request.

ZT must be applied to software, data, and systems to ensure comprehensive protection against threats. ZT verifies and secures every identity, validates device health, enforces least privilege, and captures and analyzes telemetry to better secure the digital ecosystem.

Automation is critical to a robust and sustainable ZT deployment. Routine tasks such as provisioning, access reviews, and attestation should use machine learning to defend and if necessary, restore infrastructure quickly after an attack. Given the volume of threat notifications and alerts, automation is critical to managing the digital environment at the speed and scale needed to monitor, intercept, and defeat attacks

We recommend embedding ZT into existing security layers and distributed cloud architectures. We use our ZT proving ground to test external technologies and our own solutions, our tools for Zero Trust readiness reduce adoption risks, and we are applying the latest AI/ML technology to deliver evolving, machine speed, unified network defenses. The E2C Anchor of Trust, developed in partnership with Guardtime Federal, provides digital signatures and checksums at all steps of the software development and deployment process, from code check-in/merge through build, artifact delivery, and deployment. This documents the chain of custody for third-party software, open-source software, newly developed software, and deployed software as each signature and checksum is added to the blockchain within Guardtime KSI® Calendar. KSI® Dockets provide the integrity, attribution, and provenance required from a supplier's delivery through deployment in the software development supply chain. This supports detection and forensic analysis of code injection attacks (such as SolarWinds) and will be critical to enabling continuous ATO by creating a trusted software supply chain. E2C provides an analogous capability for data provenance utilizing the Guardtime KSI Calendar blockchain. Each piece of intel data, commercial imagery, and targeting data is signed and anchored to the KSI calendar to provide a chain of custody for all the data in the ecosystem. At the operational and tactical level, data authenticity is verified against the KSI calendar.

5. Adopt a Subscription Business Model

The Army understands that constant enhancement, correction, threat mitigation, and adaptation is the rule for software and systems, not the exception: there is no "done" state, only the relentless pursuit to meet user expectations. The subscription business model shifts focus from delivery of products to provision of capabilities delivered via services. This model extends the soldier-centric design model across the entire lifecycle of a capability, from initial design through monitoring of actual use within mission threads.[12]

The Army Digital Transformation Strategy establishes the vision for how a digital transformation can help achieve Waypoint 2028, the Army's construct for Multi-Domain Operations. In that document, the Army CIO stated: "Digital transformation represents a shift in operations and culture that fundamentally changes how an organization delivers value through the adoption of advanced technologies such as cloud, data, and artificial intelligence (AI). Digital transformation is driven through innovation and new business and operating models, powered by a digital workforce that is agile, adaptive, and tech-savvy….the Army must keep pace with the rapid change in technology, adopt modern best practices, and avoid any delays from bureaucratic institutional processes."[13]

Subscriptions deliver capabilities to millions of mobile devices that are continuously updated with minimal user intervention.[14] User expectation for these devices is simple: automated configuration and updating with minimal user effort. This was not always the case, and as recently as the last decade users had to understand and work around formats and protocols if they wanted to share from one device to another. Connectivity was intermittent, file formats were proprietary, platforms were vendor-specific, and updates were burdensome and often incurred hours of troubleshooting to address incompatibilities. Vendors were in an annual race to release devices with a long list of features. Consumers expected to pay for upgrades to expand capabilities and extend functionality.

The subscription model was operationalized by companies such as Apple and Microsoft. A subscription bills customers on a recurring interval for access to products and/or services. Customers download applications (e.g., "apps") to a mobile device to provide services such as weather forecasts, photo editing, event ticketing, restaurant reservations, and more.[15] The user pays a monthly or annual subscription but does not own, manage, or control the underlying infrastructure including network, servers, operating systems, storage, or even individual application capabilities, except for limited user-specific settings. Apps are updated continually with little to no user oversight, involvement, or tasks. Major upgrades issue notifications with the ability to defer until the update will not impact current activities.

The customer's experience is the sum of all engagements with products, services, and interactions that contribute to the consumer's satisfaction with outcomes, while a failure in any one aspect reduces the overall perception of value. Consistent delivery across all domains increases the likelihood of continued subscription. This focus on experience is an evolution from transactional processes focused on buying products to a model built on relationships that meet or exceed customer expectations. Retailers and brands seek to become an indispensable part of consumers' lives through a comprehensive understanding of the customers eating, shopping, travel, work, recreation, and entertainment patterns. These patterns offer opportunities to provision value in the right place, at the right time.[16] Attaining this level of customer understanding requires a consumer-centered view of the entire business, from supply chain to content to marketing to order fulfillment to customer service to the technology ecosystem.[17]

The subscription model has direct application to the Army. Effective Continuous Modernization requires a comprehensive understanding of all aspects of a mission coupled with monitoring and analysis of threats, vulnerabilities, soldier needs, organizational constraints, and mission demands. The subscription model (Figure 1: Soldier-Centered Subscription Model) is structured to deliver capabilities to the soldier at the right place, at the right time, in a useful format. The model consists of three layers:

- **Capabilities** are the aggregations of services that soldiers employ to meet mission objectives.
- **Services** are the interfaces, processes, data, and analytics required to complete tasks, monitor conditions, or initiate new tasks.
- The **Ecosystem** consists of all technologies, data, interactions, and organizational activities that deliver services.

This approach decouples capability from a specific vendor or supplier as services and ecosystem components are modular and more readily modified, replaced, enhanced then products.

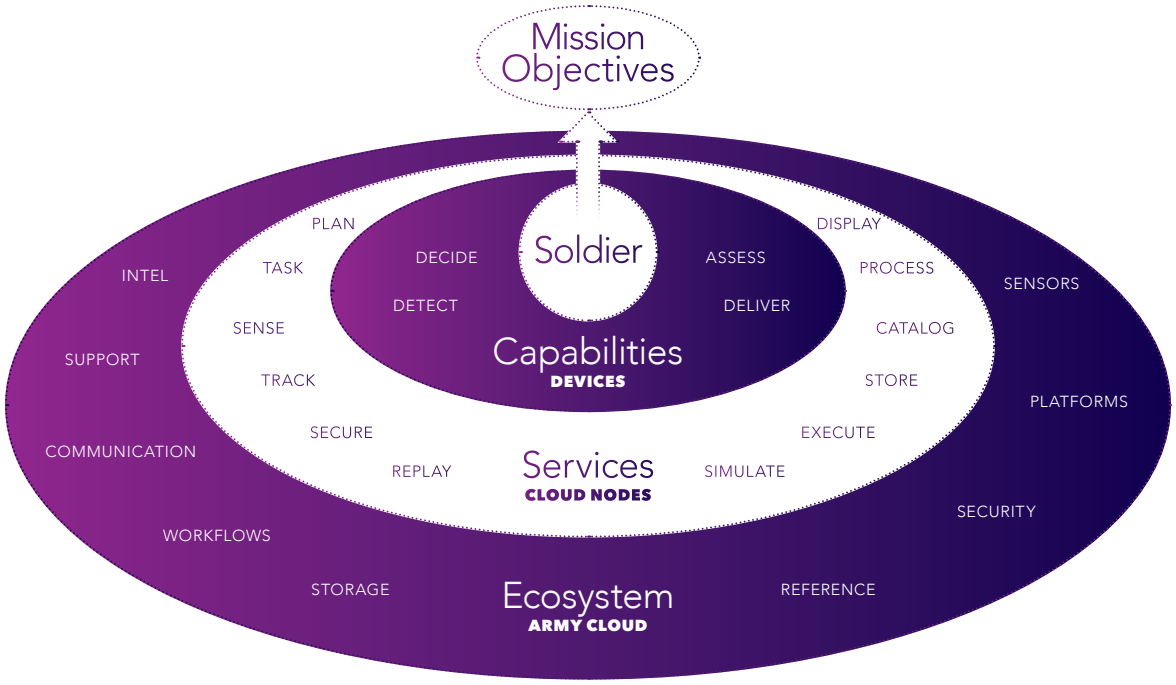

FIGURE 1: SOLDIER-CENTERED SUBSCRIPTION MODEL

The model exploits the agility and comprehensiveness of the consumer subscription model while enhancing soldier-centric design. The subscription is constructed around the soldier's mission objectives, with the entire structure focused on delivering capabilities. The subscription model does not necessarily receive all requirements from soldiers – in fact, much of the ecosystem and services that deliver capabilities are little interest to the soldier. The soldier conducting a mission is not interested in the ecosystem and services. The soldier wants capabilities that meet needs surfaced by the mission objectives. Therefore, stakeholders must understand and continually assess what the soldier needs and is likely to need and continue to extend, enhance, mitigate, and adapt the ecosystem and services to deliver those needs. The model closes the gap between requirements and point of need and provides immediate feedback of the effectiveness of the model. In addition, this approach:

- Increases Stakeholder Engagement
- Improves the Frequency and Quality of the Feedback Loop
- Increases Delivery and Provisioning Frequency
- Reduces Time to Recovery

6. Align Contracts to Continuous Modernization

Contracts built to support sequential development (i.e., waterfall) cannot adequately adapt to the efficiencies, velocity, and agility required in the information age. Continuous Modernization requires an approach to acquisition that enables rapid alignment to the operational tempo. These new capabilities must be trustworthy and resilient despite funding constraints and tight timelines, and yet be developed in such a way as to enhance agility and mission capabilities. Current contracts can accommodate Continuous Modernization within an Indefinite Delivery/Indefinite Quantity (ID/IQ) contract (See Figure 2: Hybrid Integrated in Traditional Contract Model below). This approach uses ID/IQ to frame the overall effort, Task Orders (TO) for specific Continuous Modernization work streams, and Project Task Forms (PTF) to assign discrete efforts (such as microservice development, extension, vulnerability mitigation, and adaptation). The graphic shows three software factories, however it can be as few as one or as many as can be effectively managed by the stakeholders.

This approach permits a mix of companies and/or teams scaled to the scope of the effort. This approach also enables onboarding of companies to learn the Continuous Modernization processes, methodologies, procedures, terminology, etc. The strategy also enables low-cost "bake off" sourcing where innovative companies can compete to develop a specific service with the government selecting the most capable output. Finally, the approach permits small business to participate in large scale systems development. Key to the contracting model is continuous monitoring and feedback informing each successive sprint cycle. Key to this approach is an active System of Systems scope that works with applications and development contracts to ascertain integration points and opportunities.

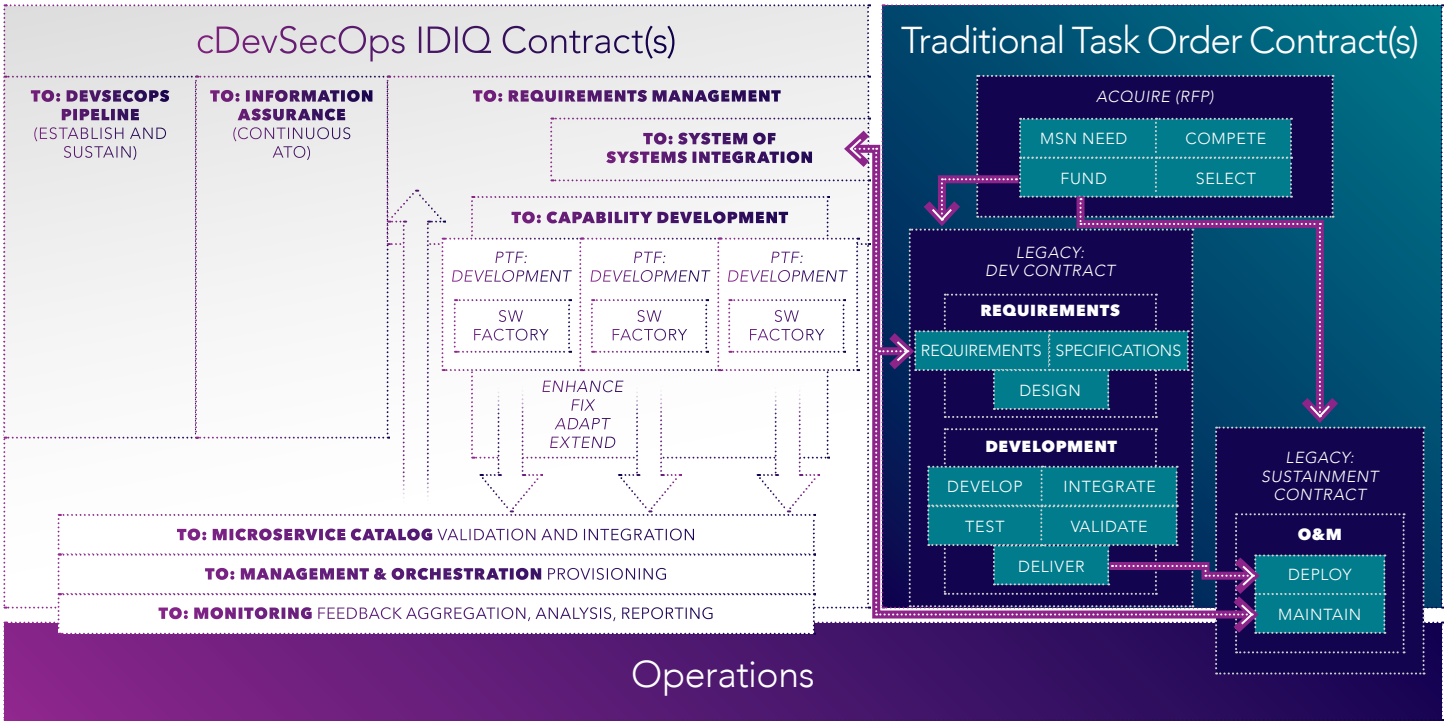

FIGURE 2: HYBRID INTEGRATED IN TRADITIONAL CONTRACT MODEL

We recommend the Army select a key mission system to demonstrate the utility and efficacy of the Continuous Modernization approach with contracts that permit agility and continuous iterations. This can be accomplished using an Indefinite Delivery/ Indefinite Quantity (ID/IQ) contract to frame the overall effort, Task Orders (TO) and Other Transactional Agreements (OTA) for specific Continuous Modernization work streams, and Project Task Forms (PTF) to assign discrete efforts (such as microservice development, extension, vulnerability mitigation, and adaptation). We also propose a comprehensive estimation process that considers all the factors that impact cost and recommend the government employ this model in evaluating estimates provided.

Implementing a Continuous Modernization approach requires a continuous partnership between stakeholders, operational users, and developers reviewing requirements, allocating LOE, and validating results. Stakeholders must have decision authority to participate in and continually direct the process. This also requires a shift in programmatic artifacts and cadence, as waterfall artifacts such as Program Management Reviews (PMR) are replaced by more frequent (and more useful) sprint events.

An incremental approach will help build confidence and familiarity in the new processes, work outputs, working arrangements, and estimates. A gradual move to cloud-based infrastructure can be facilitated by a methodical approach including repurposing legacy applications, an approach to continuous updates, a clear definition of roles and responsibilities.

1 In a 2009 Memo, the DoD CIO recognized this problem: "To effectively achieve its missions, the Department of Defense must develop and update its software-based capabilities faster than ever, to anticipate new threats and respond to continuously changing requirements." https://dodcio.defense.gov/Portals/0/Documents/FOSS/2009OSS.pdf

2 From the 2021 Army Modernization Strategy: "…the Army will continue to reform its business processes, shifting from Industrial Age to Information Age approaches, and ensuring we have sufficient funds available for the Army's modernization priorities." https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN34818-SD_08_STRATEGY_NOTE_2021-02-000-WEB-1.pdf

3 2021 Army Modernization Strategy, p2.

4 Doug Bush sworn in as Assistant Secretary of the Army for Acquisition, Logistics and Technology, OCPA, February 15, 2022. https://www.army.mil/article/254007/doug_bush_sworn_in_as_assistant_secretary_of_the_army_for_acquisition_logistics_and_technology?msclkid=19d53573cf9c11eca8fb3f873d4b33af

5 The Army Cloud Plan, 2020, https://api.army.mil/e2/c/downloads/2020/09/11/81bb912e/the-army-cloud-plan-2020-final2.pdf

6 Strangler Fig Application, Martin Fowler. https://martinfowler.com/bliki/StranglerFigApplication.html

7 For example, the current version of Microsoft® Word for Microsoft 365 MSO is Version 2201 Build 16.0.14827.20216

8 https://www.microsoft.com/en-us/microsoft-365/

9 https://www.dote.osd.mil/Portals/97/docs/TEMPGuide/DefenseAcquisitionGuidebook.pdf

10 Defense Acquisitions: How DOD Acquires Weapon Systems and Recent Efforts to Reform the Process, Congressional Research Service, May 23, 2014. https://crsreports.congress.gov/product/pdf/RL/RL34026

11 DoD Enterprise DevSecOps Strategy Guide, Version 2.1, September 2021. https://dodcio.defense.gov/Portals/0/Documents/Library/DoD%20Enterprise%20DevSecOps%20Strategy%20Guide_DoD-CIO_20211019.pdf

12 A "mission thread" is defined as what the soldier needs to effectively complete the actions required to deliver a specific mission result. "A mission thread is a sequence of end-to-end activities and events presented as a series of steps that accomplish the execution of one or more capabilities that the SoS supports." An Introduction to the Mission Thread Workshop, Carnegie Mellon Software Engineering Center, March 2015. https://insights.sei.cmu.edu/blog/an-introduction-to-the-mission-thread-workshop/

13 Army Digital Transformation Strategy, HQ Department of the Army, October, 2021. https://api.army.mil/e2/c/downloads/2021/10/20/3b64248b/army-digital-transformation-strategy.pdf

14 By 2016 Apple, Inc. had sold over 1 billion iPhones, https://www.theverge.com/2016/7/27/12302542/apple-billion-iphones-sold

15 There are over 3.5 million apps available on the Apple App Store: (https://www.statista.com/statistics/268251/number-of-apps-in-the-itunes-app-store-since-2008/?msclkid=b25de542ce7711ecaba6de6438e2c917) and 2.5 million on the Android Google Play (https://www.appbrain.com/stats/number-of-android-apps?msclkid=e2e9720ace7711ec931f917f37137ae7)

16 Product Centric vs Customer Centric: A Continuous Retail Evolution, https://blog.crobox.com/article/product-centric-vs-customer-centric

17 6 Ways to Build a Customer-Centric Culture, Harvard Business Review, https://hbr.org/2018/10/6-ways-to-build-a-customer-centric-culture?msclkid=8d4c5900ce7811ecab3776c7f85575d6

1750 PRESIDENTS STREET, RESTON, VA 20190