



Enabling Software Innovation Across the DoD

Jeremiah Robertson ■ Trey Bonner

THE GROWING CRITICALITY OF SOFTWARE across the Department of Defense (DoD) coupled with the progressive decline in software development performance and capability across prime defense contractors presents a critical need for a new strategy for software development and acquisition. Current development is plagued by schedule delays, cost overruns, and requirements creep due to long acquisition cycles and shifting priorities. Agencies across the DoD have highlighted this deficiency while the threat from near-peer adversaries continues to evolve rapidly, and major systems increasingly depend on software to maintain a technological advantage.

Robertson is a consultant with 4 years of experience in providing management, simulation/modeling, and technical support for major Department of Defense and Intelligence Community acquisition programs. **Bonner** has 6 years of experience in developing and implementing successful operational strategies in highly complex environments from Development Operations transformations to factory flow optimization.



The March 2019 Defense Innovation Board report, “Software Is Never Done: Refactoring the Acquisition Code for Competitive Advantage,” stated:

The need to meet future threats requires us to rethink how we develop, procure, assure, deploy, and continuously improve software. DoD’s current procurement processes treat software programs like hardware programs, but DoD can no longer take years to develop software for its major systems. Software cannot be an afterthought to hardware, and it cannot be acquired, developed, and managed like hardware. DoD’s acquisition and development approaches are increasingly antiquated and do not meet the timely demands of its missions.

A new strategy is needed to meet DoD objectives while practically merging principles from commercial software development and government acquisition realities. Development, Security and Operations (DevSecOps) and Agile have emerged as the preeminent strategies for developing

higher-quality software in shorter timespans while lowering costs.

Numerous companies implemented an Agile or DevSecOps-like framework with results showing improved performance in development at a minimal additional cost. These success stories are great examples of identifying waste within a value stream and eliminating it to focus on the products and processes that provide the maximum business value to clients. However, Agile and DevSecOps worked for these companies for one extremely important reason: They identified the problems first and developed a specific process for overcoming them. After figuring out that they were developing too many software releases, they focused on a regular cadence. They realized that developers on different teams weren’t communicating about dependencies and risks and then decided to incentivize communications among teams. Rather than implement new processes to continue addressing the symptoms, they identified the root causes to solve the problems “once and

for all.” These companies weren’t successful because they implemented Agile or DevSecOps in name only; they were successful because they identified their problems and solved them by adhering to principles that provided practical solutions for overcoming their problems.

What’s the Problem?

Nelson Repenning in 2016 shared a relevant story in “What Problem Are You Trying to Solve: An Introduction to Structured Problem Solving,” published by the MIT Sloan School of Management. In the early 1990s, the manager of one of Harley-Davidson’s engine manufacturing and assembly plants, Don Kieffer, hired Hajime Oba (the architect behind the famous Toyota production system) as a consultant. Kieffer wanted to implement the Toyota production system in his plant and believed Oba would be the perfect person to help do it. However, to his surprise, Oba told him to fix a problem in one specific area. Kieffer was upset, and Oba asked him, “What problem are you trying to solve?” Don replied, “The problem I’m trying to solve is that I want to implement the Toyota production system, and I don’t know how.” Oba replied, “If you want to implement the Toyota production system, tell me what problem you are trying to solve.” The day did not end well, but the lesson from Oba is very clear: Kieffer assumed a solution (the Toyota production system) without first understanding what problem he was trying to solve.

Unfortunately, DoD may be in the same predicament. There is a widespread assumption that Agile and DevSecOps will produce high-quality software cheaper and faster than any other method. However, unless the root problems are directly addressed, DoD is implementing a solution and strategy that not only may fail to address the problems, but could make current software production worse as they address the symptoms rather than the cause. DevSecOps and Agile development processes were developed in order to adapt and capitalize on particular customer behaviors across the commercial markets in which they operate and capitalize on. In order to reach the DoD’s objectives, a methodology must be specially tailored with specific processes for how software will be developed within the DoD acquisition and usage framework.

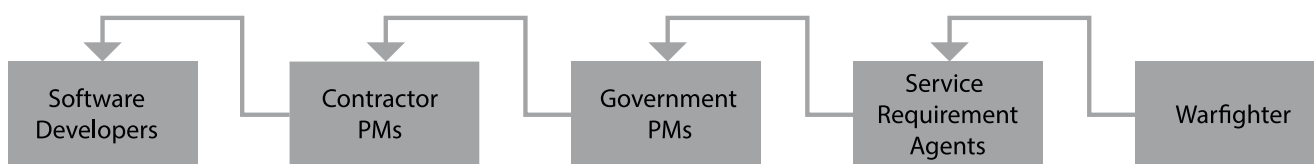
Problems Across DoD Software Development

Software development and acquisition face numerous challenges inherent in the DoD acquisition and operational framework. Integrating strategic planning and analysis of warfighter needs with executional realities will help solve the problems and not just address the symptoms.

Requirements churn is one fundamental problem that will require innovative approaches for more rapid delivery of capabilities to the warfighter. Hard-and-fast requirements have resulted in development focusing on completing requirements like a checklist instead of innovating the best approaches for providing value to the customer. Instead of having a strict set of capabilities that must be developed, the question needs to be continuously asked: What does the end user need the software to accomplish? Constantly asking this question enables innovation as it opens the aperture for the realm of the possible. This also produces an environment for funding the unknowable requirements of the future. The shift from detailed and firm requirements to requiring activities (i.e., describing the contextualized needs of the warfighter) will enable software development teams to produce highly effective software that more accurately meets true needs by eliminating the requirements flow as shown in Figure 1.

Each step within this process diminishes the potential for software developers to meet the warfighter’s real needs. To prevent this degradation, DoD must streamline the flow of required activities to software developers. A greater number of process steps between the product developer and the user leads to a greater dilution of true requirements and limitations on innovations, requiring additional rework with a longer correction cycle time. Customer awareness of the cycle time and potential impacts of changing software needs can minimize the risks of too much iteration. Excessive iteration risks continuous churn of required activities and ongoing shifts in priorities as people keep changing their minds. This can result in shorter development cycle times and prove slower, costlier or more disruptive. Therefore, the customer must

Figure 1. Shift From Detailed and Firm Requirements to Requiring Activities



Key: PMs=Program Managers

Source of figures: The authors.

...unless the root problems are directly addressed, DoD is implementing a solution and strategy that not only may fail to address the problems, but could make current software production worse as they address the symptoms rather than the cause.

be aware of the cycle time and the impacts of changing software needs.

The pace of innovation of software development in the commercial sector now is faster than the requirements development and decomposition cycle. The long-range planning of requirements is not conducive to quick deployment of cutting-edge technology that is developed at a much faster pace. This means that adversaries with more centralized decision-making processes than those of the DoD could more effectively leverage this pace to drastically reduce the time to field capabilities and thereby erode the DoD's technological advantage. This requires new strategies and approaches for developing or acquiring software. The DoD must manage the development of new software to meet its objectives. This would require that the DoD become a highly skilled practitioner of various software development approaches and invest in the proper tools, processes, and people to cultivate innovation and bring the developer closer to the end user.

Develop or Acquire: Different Strategies

The DoD must establish a vision for future software investment and become a practitioner of DevSecOps. Investments will be needed in the proper tools, processes, training, and technology enablers to build the ability to innovate as fast as the commercial sector. For most current programs, this simply is not an option because of the program's magnitude, the relationship with current contractors, and the high stakes. In the future, the decision to create rather than buy must be made when developing program strategies and that decision will fundamentally dictate the operational framework for meeting DoD objectives.

In some software use cases, the DoD must be the developer as DoD builds and maintains the operating environment where platforms must integrate. This will be required to execute future strategies such as the Next Generation Air Dominance (NGAD) strategy and to ensure that hardware can innovate at the same rate as software across multiple contractors and vehicles. The NGAD strategy is paramount to reducing the cost and bureaucratic ineffi-

ciencies that have plagued the defense industry and every major program over the last 40 years, but will heavily rely on the U.S. Air Force (USAF) strategy for acquiring software. If multiple contractors are not able to work within the same development environment where USAF manages the software baseline and technical integration, NGAD will result in numerous platforms with limited interoperability and will ultimately fall short of its objectives.

To effectively acquire software, on the other hand, a strategic approach requires the effective management of a contractor, or many contractors, to meet DoD objectives. The framework that enables this approach relies on the right acquisition strategy rather than the contractor's existing software development methodology. Acquiring software generally requires an acquisition authority that comes between the user and developer of the solutions with additional contracting requirements. This separation is inherently not Agile and requires an innovative strategy to meet DoD objectives rather than a silver bullet solution.

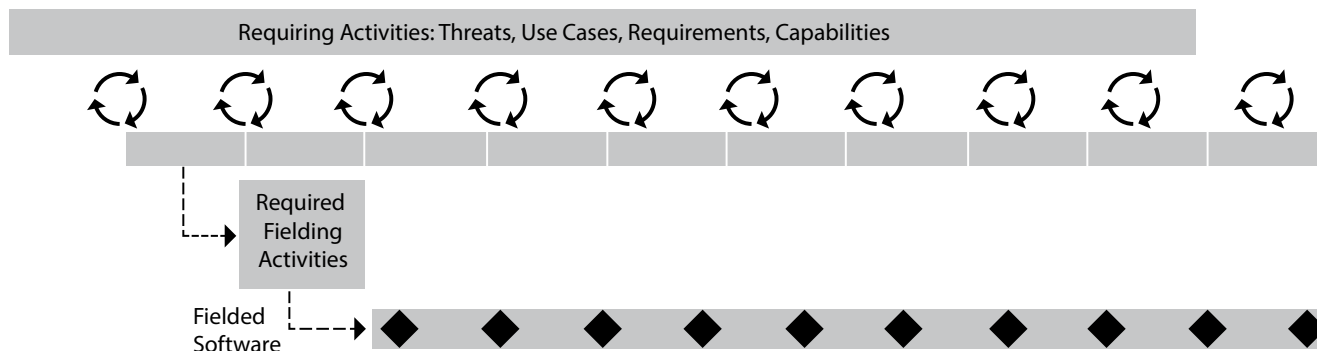
Future Objectives, Strategies and Enablers

A strategic approach to implementing a specific development strategy (whether Waterfall, Agile or DevSecOps) through traditional contract relationships with industry while accelerating capability to the warfighter creates numerous issues throughout the software development and testing cycle. Critical enablers must be established before determining the best contracting strategy. First, a program must have an open, free flow of capital and resources. This would alleviate contracting and bureaucratic delays while maintaining accountability for the affordable, timely and predictable delivery of requirements and capabilities. Second, there are technological requirements that must be considered. To produce regular releases, specific hardware and software are needed for automated builds and to perform automated testing. Without the right tools, releases are difficult to produce, test and distribute to the field.

The bespoke or specially tailored acquisition strategy for effectively managing contractor behaviors relies on a mix of contracting activities that meet the needs of both the

Figure 2. Acquisition Framework That Meets Enablers and Can Deliver on DoD Objectives

Indefinite Delivery, Indefinite Quantity for Planning and Management of Requiring Activities



Software Factory Contract:

- Multi-year, Firm Fixed Price Contract with fixed cost per fielded release.
- Include incentives for release accuracy, new functionality, and efficiencies.
- Use a Production Non-Recurring allocation for capital requirements to ensure efficient use of funding.

DoD and the contractor's corporate interests. To achieve DoD objectives, the acquisition strategy must account for all these enablers:

- Provide open, free flow of capital and resources to alleviate contracting and bureaucratic delays.
- Maintain accountability for the affordable delivery of requirements/required activities.
- Provide predictability for capability delivery.

A "Software Factory" approach allows the DoD to align objectives and manage contractor behaviors by assessing the various stages of the software development cycle and incentivizing the contractor when bottlenecks or inefficient behaviors become evident. Figure 2 depicts the acquisition framework that meets these enablers and can deliver on the DoD's objectives.

By assessing the software "factory" the DoD can gain insight and measure specific stages within the development cycle. In the same fashion that a traditional factory would be assessed, program managers are able to determine the capacity currently available to produce new software, where any bottlenecks occur, and potential investment in new tools or processes to increase the capacity. This is accomplished by carefully measuring the right performance areas across the development cycle and integrating them into a complete system. By leveraging data analytics and visualization, program managers are able to quickly see why release content is behind schedule or where teams are performing poorly. This allows program managers to devise incentive strategies to encourage innovation and decrease the time necessary to field new solutions.

Funding contractors by release provides a stable, predictable measure for cost over the course of the program, which allows for the use of longer-term contracts. Each release is firm fixed price (FFP); however, the content is based on specific performance criteria with incentives for delivering capability to the warfighter. Measures such as release accuracy versus plan, backlog cycle time of new features, or use-case fulfillment can be used to incentivize the contractor while ensuring that the DoD is getting the software needed. This activity maintains accountability for the affordable delivery of required activities while providing the long-range, free flow of the capital that contractors need to invest in performance improvements.

Other factors also must be examined, such as data rights and ownership of source code. The lack of ownership in source code is the greatest factor currently limiting DoD's ability to implement commercial best practices and leverage competition against incumbents on large programs. There is some additional concern about the organic depot and contractor dynamic. The DoD must maintain a future ability to support software dependent platforms when contractors do not provide the necessary support or modernization as they move on to new programs. DoD must develop a strategy that incentivizes innovation by the contractor as well as collaboration with software depot teams to organically develop capabilities across the Services.

Enabling the Future of Software Innovation

DevSecOps provides an exciting opportunity for the DoD to leverage principles that have been proven sound in the

... unless the root problems are directly addressed, DoD is implementing a solution and strategy that not only may fail to address the problems, but could make current software production worse as they address the symptoms rather than the cause.

commercial industry as ways to foster innovation. However, the success of future DoD software development programs more heavily depends on accurately setting objectives, confidently choosing between a make-or-buy path, and then developing a strategy for meeting those objectives. While DevSecOps provides excellent principles for improving DoD software development, implementing it without considering the DoD's execution realities can introduce more program risk than reward. Software development across the DoD will be enabled by the open, free flow of capital and resources to alleviate contracting

and bureaucratic delays, the ability to maintain accountability for the affordable delivery of requirements/required activities, and predictability for capability delivery. By fulfilling the enablers for fostering software innovation and developing models for providing predictability into the software "factory," program managers can more rapidly meet program objectives, reduce time to field new capabilities, and improve the capability delivered to the warfighter.

The authors can be contacted at Jeremiah.Robertson@pausinc.com and Trey.Bonner@pausinc.com.



DAU ALUMNI ASSOCIATION

Join the Success Network!

The DAU Alumni Association opens the door to a worldwide network of Defense Acquisition University graduates, faculty, staff members and defense industry representatives—all ready to share their expertise with you and benefit from yours. Be part of a two-way exchange of information with other acquisition professionals.

- Stay connected to DAU and link to other professional organizations.
- Keep up to date on evolving defense acquisition policies and developments through DAUAA newsletters and the DAUAA LinkedIn Group.
- Attend the DAU Annual Acquisition Training Symposium and bi-monthly hot topic training forums—both supported by the DAUAA—and earn Continuous Learning Points toward DoD continuing education requirements.

Membership is open to all DAU graduates, faculty, staff, and defense industry members. It's easy to join right from the DAUAA Web Site at www.dauaa.org, or scan the QR code:



For more information call 703-960-6802 or 800-755-8805, or e-mail dauaa2@aol.com.