## **STARTING SIMPLE WITH JADC2**

By Dr. Jordan L. Fletcher and Dr. Eliahu (Eli) H. Niewood



# "

A complex system designed from scratch never works and cannot be made to work. You have to start over, beginning with a working simple system.

John Gall, as referenced in a U.S. Air Force Command and Control briefing

## Abstract

This article explores how agile systems engineering (SE) and a simplified approach can augment or complement traditional SE activities for a complex system of systems. The Department of Defense (DoD) Joint All Domain Command and Control effort is identified as a potential candidate to apply agile SE toward, given the number of SE challenges identified during MITRE's 2022 independent assessment. Taking an agile, federated, and "bottom-up" approach does not require abandoning rigor in systems engineering. Rather, that rigor should be applied to meeting user needs, ensuring working products, and rapidly iterating to fix issues and extend capabilities. Architectures, standards, and interface controls should not be abandoned, but they should be simplified where possible and "grow" into a working minimum viable product. A key point made in this article is that a bias for action must be undertaken to start somewhere, even if it is not the ideal starting point, so that the DoD can experiment, learn, adapt, and iterate toward a working solution from a simple beginning.

## **Starting Simple**

The Department of Defense (DoD) seems to perennially struggle with implementing complex systems or systems of systems. Those struggles are widespread enough that it is surprising and noteworthy when a large program maintains its schedule or is fielded within its budget, as evidenced by public discourse surrounding the new B-21 bomber. More common than the successes are those programs that arrive over budget and behind schedule, or those that never arrive at all. The replacement for the Air Operations Center (AOC)—also known as the AOC 10.2 program—was canceled after failing to meet budget or schedule even after multiple revisions. A new system to replace the Space Domain Awareness Capability (SPADOC) has been close to delivery for years, having undergone multiple changes in name without yet fielding actual capability, preventing decommissioning of the legacy SPADOC system.

The Next-Generation Global Positioning System (GPS) Operational Control System (OCX) is critical to enabling new capability already on GPS satellites in orbit today, and yet OCX continues to lag its frequently revised schedule.

Often systems engineering receives the lion's share of fault for these delays, overruns, and cancellations. "If the program had just done a better job of following the systems engineering 'Vee' the program would have been on time," said every traditional systems engineer, ever. New systems engineering methods, like digital engineering representations of complex systems and their interactions in tools like Magic System of Systems Architect/Cameo, are seen as a means of avoiding future problems through managing complexity. Increasing the resourcing dedicated to systems engineering activities and performing better systems engineering up front are often cited as ways to avoid future problems. But what if the problem is more fundamental? What if, as Gall wrote, complex systems cannot be designed from scratch but must start with a simple working system and then be allowed to grow in complexity from there? A savvy reader might note that AOC 10.2, the SPADOC replacement, and GPS/OCX were all attempts to build new systems as complex as what their predecessors had evolved to over time. In these cases, maybe we need to follow the advice in Dan Ward's simplicity cycle and take an iteration to simplify the existing system, instead of layering on new requirements and features:

We explore the practice of creation-throughsubtraction, rather than continuing down the path of creation-through-addition ... Upon arriving at the peak of complexity, we set aside, at least for the moment, our additive, expansive techniques and turn instead to the sculptor's toolbox.<sup>1</sup>

Our hypothesis is then that <u>a successfully</u> <u>engineered complex system should start with</u> <u>a simple working system, and then grow in</u> <u>complexity</u>. We have to account for the point at which a system has reached a limit of complexity and find ways to simplify as we evolve or modernize. Also, when confronted with an existing complex system that needs to be replaced, we need to find ways to break the problem into smaller, simpler pieces that can be successfully resolved.

One place where this "start simple" approach might prove worth trying is the set of systems needed to enable what is known as Joint All Domain Command and Control (JADC2). JADC2 isn't a single program; it was defined by the Joint Requirements Oversight Council (JROC) as a set of enabling capabilities that need to be brought together operationally and employed by Joint and Combined forces. JADC2 was introduced as Multi Domain Command and Control by the U.S. Air Force and Multi Domain Battle by the U.S. Army in 2016–2017. By merging the two concepts into JADC2 and tasking the Joint Staff J6 with oversight, the JROC sought to ensure the multidomain capabilities and concepts included all Services and all warfighting domains. To better understand what can be gained by "starting simple" with JADC2, it is worth reviewing the state of the effort in 2022, how it has improved over the past year, and what can be done to continue making strides in the right direction.

## **Assessing JADC2**

In early 2022, MITRE performed an internally funded independent assessment looking at the progress made by the DoD at implementing JADC2 capabilities. The landscape has certainly evolved over the past year, but the early-2022 assessment concluded that the DoD was not on a path to success with JADC2, and that it was unlikely the DoD would achieve its desired end state with JADC2 given the strategic direction and governance construct in place at the time. This conclusion was based on a number of factors, including that the goals for JADC2 weren't clear or actionable, that a number of the documents written to guide JADC2 were cumbersome and lacked coherence, and that although each Military Service was performing laudable work on modernizing its Command and Control (C2) systems, they weren't aligned to a common JADC2 approach.

In response to the independent assessment, MITRE developed a set of recommendations for how the DoD could accelerate the fielding of JADC2 capabilities. These recommendations, included below, echo some of the same points made for starting simple:



#### DEFINE A FOCUS FOR JADC2.

MITRE recommended developing initial JADC2 capabilities to support deep sensing and long-range effects against peer adversaries in challenging geographical environments. The intent was to provide a specific target set with a real mission context. This focus would provide JADC2 capability developers with sufficient context to understand which problems they need to address and which ones they don't. It also enables the use of operational analysis to assess the utility of better C2 operational metrics. This analysis could lead to meaningful goals for decisionmaking time and weapon optimization, for example, and allow acquisition officials to better prioritize their resources.



#### ADOPT A FEDERATED APPROACH TO

JADC2. Instead of trying to build one large complex system through a joint effort or within a single program office, have the Services continue their work to develop the systems they each need to provide C2 of their platforms, weapons, and sensors, such as route planning and communications scheduling. JADC2 development could then be focused on how to knit capabilities across those Service systems in a loosely coupled manner. Federated all-domain situational awareness entails building needed understanding using information provided by Service and Intelligence Community sensors rather than trying to flow all data from all sensors to everyone. Federated tasking means intelligently allocating platforms, sensors, and weapons to the various sensing and effects tasks required from an all-capability, all-Service perspective, but then allowing each Service to implement tasks in Service-specific ways.



#### BUILD JADC2 FROM THE BOTTOM UP.

Traditional systems engineering has been leading toward a top-down effort to implement JADC2 capabilities, whereas MITRE recommended a bottom-up approach of iterative experimentation to determine what really needs to be built. A top-down effort will define a large and complex architecture using all the requirements and use cases for JADC2, applying detailed digital engineering to build the needed capabilities. By contrast, the DoD should stitch together a minimum viable product (MVP) from existing efforts and capabilities that will provide an initial level of effectiveness against the specific mission focus defined for JADC2. Over time, the DoD can spiral in advanced technology and new capabilities to replace specific elements of the MVP or to add elements addressing new or expanded mission needs.



#### USE EXPERIMENTATION TO UNDERSTAND

GAPS. Assuming development of a JADC2 system of systems built from existing capabilities, the DoD could simultaneously assess the most impactful gaps and begin development of the capabilities needed to address those gaps. Continuous and iterative experimentation with the MVP would assist with gap assessment while ensuring Joint operators have usable capability at any point in time. These perceived gaps exist not just across operational capabilities, but also in doctrine and in operational organizational structures. Where gaps are insurmountable or difficult to address, further development of the bottom-up prototype may help get around the gaps and institutional barriers.

#### **EXPAND THE USE OF OPERATIONAL**

**ANALYSIS.** Operational analysis can be used to support JADC2 efforts in multiple ways. As mentioned previously, it can help show the mission value resulting from improvements in situational awareness and decision making. It can also provide a means for showing how different commercial industry concepts and capabilities would impact operations in the chosen mission focus for JADC2. An operational analysis sandbox could allow concepts from across commercial industry to be brought together to realize synergies. This activity supports the experimentation-focused approach and enables the ability to decompose a complex problem into simpler, smaller problems.

Starting in June 2022, these recommendations were socialized widely with elements of the DoD, particularly organizations in the Office of the Secretary of Defense. In general, DoD leadership supported the recommendations, was actively working on some related efforts, and expressed interest in acting on other recommendations. At the same time, MITRE worked with multiple government offices developing specific systems and capabilities that could be part of a JADC2 MVP associated with a chosen mission and target set. A number of those organizations are now working together to field an initial operational prototype with leave-behind capability as part of experimentation in the 2023–2025 timeframe. If successful, this will be one of the first activities to provide operational units with capability that starts to meet the promise of JADC2, and it will have started from the bottom up.

In January 2023, the Government Accountability Office (GAO) developed a report (GAO-23-105495) addressing, among other topics, DoD progress toward JADC2. The GAO report made similar observations as the MITRE study, noting that no capabilities have been delivered to the Joint warfighter, and that there is a lack of detail around goal and capability alignment despite a significant amount of progress in plan development. A congressionally mandated report will result in an inventory of JADC2-related development efforts, objectives, costs, and schedules, as well as JADC2 capability gaps and performance goals. Taking action on that information is the next most important step.

## **Motivating a Bias for Action**

While efforts to build JADC2 "from the bottom up" have not yet demonstrated complete success, they have at least started to provide something tangible, leveraging operational capabilities from multiple Services and warfighting domains. MITRE's JADC2 recommendations propose a different approach than traditional systems engineering, to recognize that at some level of complexity there may be no first principles design methodology that can fully account for the complexity needed. Moreso, the recommendations are an acknowledgment that JADC2 won't happen overnight, and it will be necessary for the DoD to adjust course due to budget, technology, and operational realities. The military operational and threat environments—including policies and organizations—are variable, and thus even a perfect top-down plan will have unpredicted results over the time it takes to implement JADC2. This is a different approach than starting from scratch and perfecting the systems engineering template. It is also a different approach than up-front traditional systems engineering, where one would use a tool to design a highly detailed model for all JADC2 functions and every interaction.

As a case in point of why this bottom-up approach might have a chance of success, we can turn to the program that replaced the AOC 10.2 effort. AOC Pathfinder, which was later named Kessel Run, took the complex problem of AOC modernization and applied modern software development methods to deliver working software to a specific AOC in ways that allow continuous development and delivery of warfighting capability. The Kessel Run team demonstrated the ability to field operational capability in less than a year by putting aside traditional approaches to acquisition and systems engineering, starting with a focus area that was a subset of the problem (Tanker planning), engaging their end users, and simplifying the MVP so that refuel scheduling was basically like solving a math problem. While the Kessel Run team still has a long way to go before they modernize all of the AOC functions, they are churning out software on a daily basis that warfighters are actively using. They are also on a path to deprecate the legacy capabilities for one AOC as they field the modernized capability. This example shows how the DoD can apply flexibility and agility to deliver capabilities on budget and ahead of schedule.

Deciding exactly which JADC2 capabilities the DoD should deliver, and in what order, will always be a topic of debate. There continue to be discussions for prioritizing communications (e.g., networks), situational awareness systems (e.g., common operating pictures), artificial intelligence systems—even discussions of building JADC2 foundations first through identity and data management systems. Rather than getting stuck in analysis paralysis, the DoD JADC2 community can learn from Kessel Run's initial success and become less focused on where they start, and instead just get started. There are few airmen who would say Tanker scheduling is the most important mission of the AOC. But not only has that mission had an overwhelmingly positive return on investment; it has resulted in a groundswell behind the rest of the AOC modernization effort. Kessel Run calls this "a bias toward action," which is based on Colin Powell's 40/70 rule that states leaders need between 40 and 70 percent of the information to make a decision. Less than 40 percent means they are likely to make a mistake, and waiting for more than 70 percent means they will likely be late. If the DoD stalls to deliberate what should be built first, it risks having nothing at all when it is needed most. The JADC2 community should adopt this same bias for action-agree to build something collectively that has mission merit and move on from there. That adoption of a bias for action will be the first step in changing the narrative from JADC2 as an unclear and aspirational idea into the narrative of JADC2 being the warfighting engine behind our Joint and Combined forces.

## **Applying Agile Systems Engineering**

Perhaps it is time to recognize the limitations of traditional systems engineering alone to design and guide development of highly complex and integrated systems. Just as DevOps and agile software development have introduced a less linear approach to software development, maybe it is time to recognize that the linear, traditional systems engineering "Vee" model moving from concepts to requirements to capability development isn't effective for highly complex systems that must continually adapt to an evolving threat and operational need. What is the point in developing testable requirements against a threat that will evolve in the years before the capability is fielded? While traditional systems engineering might be relevant and needed for satellites or airplanes where a modular and iterative approach won't work (What does an MVP for an airplane look like? No engine and one wing?), it is less needed and less feasible for a software-focused system or loosely coupled set of hardware elements. Instead, it might be more effective to apply agile systems engineering principles against the problem set.

In a DoD context, agile systems engineering involves iterative capability development starting with relatively simple goals and requirements against narrowly defined mission sets. In fact, this idea of starting small and simple pervades almost every concept in agile systems engineering. MITRE's acquisition in the digital age provides a prime example:

The central guideline in developing and maintaining architectures in an agile environment is to keep it simple ... The value of the architecture decreases as the complexity increases, with volumes of documents and artifacts making it difficult for participants to comprehend.<sup>2</sup>

Beyond this goal of starting (and keeping) simple, most agile systems engineering efforts abide by the same core principles of the Agile manifesto:

- · Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan<sup>3</sup>

Applying agile systems engineering is not a panacea for all the issues surrounding complex DoD system acquisitions. In fact, it introduces just as much risk as it mitigates. However, with agile systems engineering, we can fail fast and fail often with fewer long-term consequences because we have not incurred the same programmatic and technical debt as traditional systems engineering. Some development efforts and experiments that start with the limited, simplistic requirements will fail. Some efforts will be successful but will be "dead ends" unable to scale to take on a larger mission set. Some efforts, however, will both succeed and demonstrate the ability to scale past their simple beginnings. Through the iterative and adaptive nature of agile systems engineering, we can explore all possibilities. Starting from the bottom up, with an agile mindset and a bias for action, may be the only way to

consistently find success in development of software-based systems and capabilities like those needed for JADC2 and other capabilities critical to our national security.

## **About the Authors**

**Dr. Jordan L. Fletcher** is a department chief engineer in the MITRE National Security Engineering Center. Jordan joined MITRE in 2007 following military service as a U.S. Air Force officer, and has focused his career on command, control, communications, coalition systems, and software modernization. He has supported Air Force, Army, Joint, Intelligence, Homeland Security, and International programs in the United States, Europe, Africa, the Indo-Pacific, and the Middle East. His current technical focus areas are autonomous systems and agile systems engineering.

**Dr. Eliahu (Eli) H. Niewood** is vice president, Air and Space Forces, in MITRE's National Security Sector. He previously was vice president, Cross-cutting and Intelligence, leading MITRE's efforts addressing joint and multi-agency solutions. He joined MITRE in 2017 after serving as a technical adviser to the U.S. Air Force Rapid Capabilities Office director. Eli also served as head of the Engineering Division at MIT Lincoln Laboratory, where he oversaw development of space payloads and aircraft sensor systems. From 2011 to 2014, Eli chaired the Air Force Scientific Advisory Board.

## Citations

<sup>1</sup> Dan Ward, The Simplicity Cycle, 2015. [Online]. Available: <u>http://www.thedanward.com/the-simplicity-cycle/</u>

<sup>2</sup> MITRE, "Acquisition in the digital age – agile systems engineering." [Online]. Available: <u>https://aida.mitre.org/agile/</u> <u>agile-systems-engineering/</u>

<sup>3</sup> K. Beck et al., "Manifesto for Agile software development," 2001. [Online]. Available: <u>https://agilemanifesto.org</u>

MITRE's mission-driven teams are dedicated to solving problems for a safer world. Through our public-private partnerships and federally funded R&D centers, we work across government and in partnership with industry to tackle challenges to the safety, stability, and well-being of our nation.

#### MITRE | SOLVING PROBLEMS FOR A SAFER WORLD