CHALLENGES OF ADOPTING DEVOPS FOR THE COMBAT SYSTEMS DEVELOPMENT ENVIRONMENT

LT Andrew W. Miller, USN, Ronald E. Giachetti, and Douglas L. Van Bossuyt

The Department of Defense (DoD) is often exhorted to adopt best practices from industry, and more recently, innovation in software development as exemplified by Silicon Valley. Yet, the DoD is vastly different from industry in multiple aspects, and adoption of such practices is not as straightforward as in industry. This article investigates the challenges of adopting Development and Operations (DevOps) in the U.S. Navy for combat systems. The authors conducted interviews of multiple subject matter experts in the Navy and DoD familiar with software development, DevOps, and the DoD's acquisition processes. The observations collected from the interviews were organized and classified into either organizational, process, regulatory challenges, and technical challenges. The majority of the challenges cited were nontechnical challenges dealing with regulations, organization culture, and process. Knowledge of the challenges could help acquisition leaders in planning for, and adapting DevOps to, the Navy's acquisition process to improve DoD's software development and maintenance processes.

DOI: https://doi.org/10.22594/dau.21-870.29.01
Keywords: *Software Acquisition, Agile Methods, DevSecOps, Change Management, DevOps*

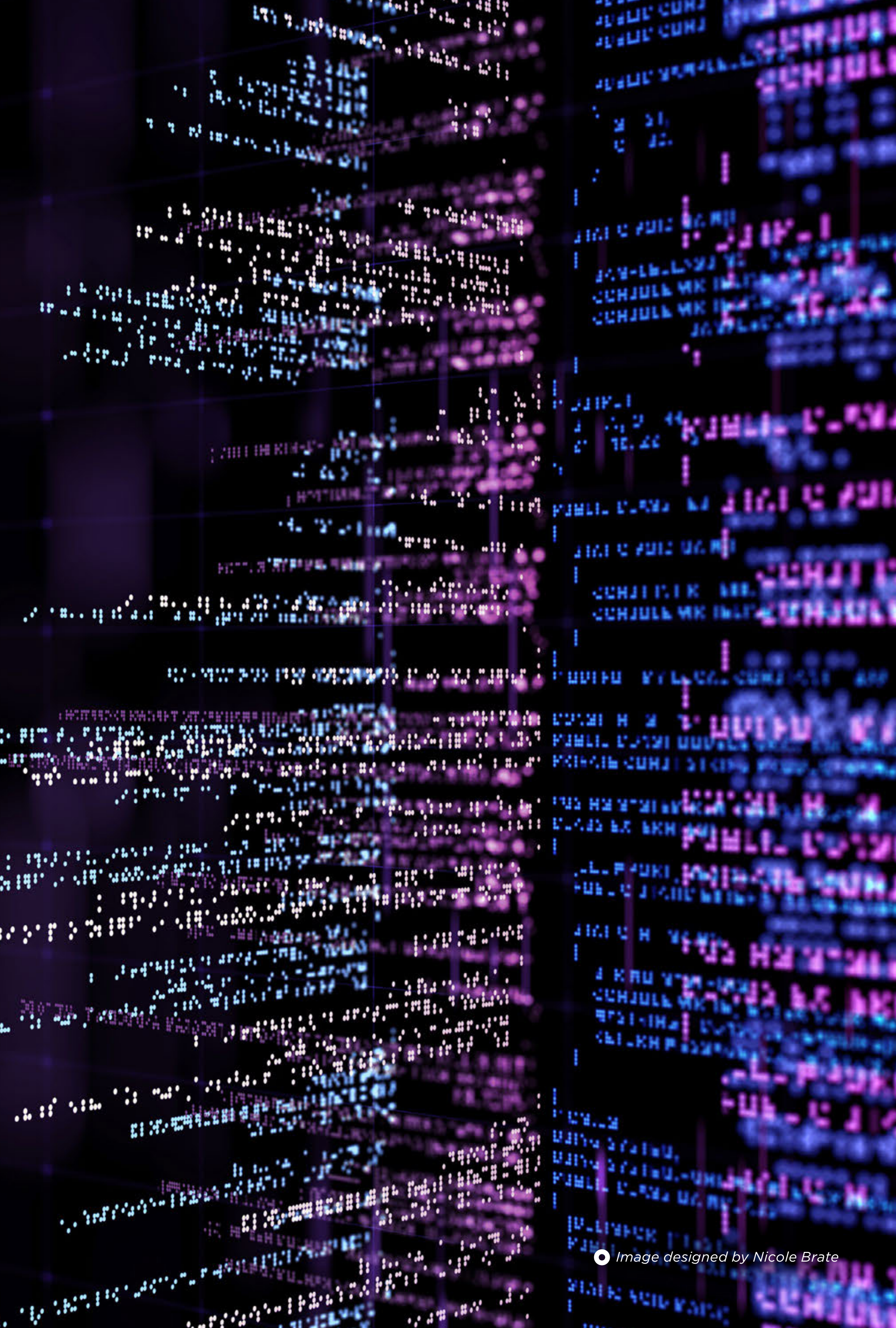
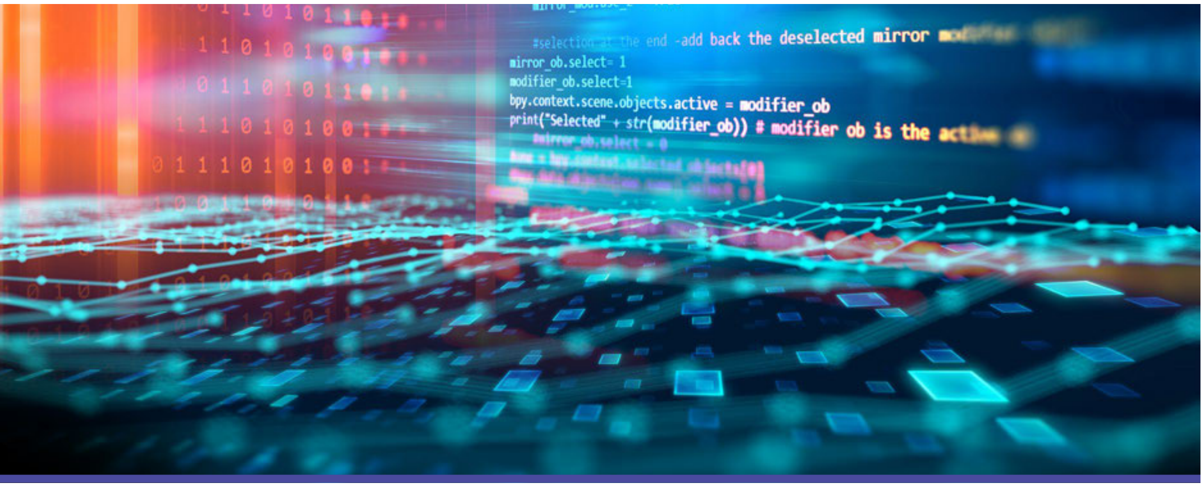


Image designed by Nicole Brate

The U.S. Department of Defense (DoD) acquisition process, system development process, and organizational mindset all evolved based on the design, development, and delivery of hardware systems. Yet, the software content of systems is increasing dramatically. Software differs signifcantly from hardware in important characteristics. Unlike hardware, software systems are intangible and are not subject to the laws of physics (Sommerville, 2021). This makes software easily deployable and updated with only a network connection. Consequently, software companies routinely push software updates to their customers, resulting in a continuous process of software development and deployment. One of the largest distinctions between hardware and software is the fact that software must be constantly updated by developers throughout its service life. In fact, the DoD finds that software maintenance, consisting of modifying and updating software to stay abreast of evolving operational needs, accounts for the majority of software budgets, and sailors commonly complain about slowness in updating software (McQuade et al., 2019). Industry has long recognized how software differs from hardware, and it uses different processes for software development than hardware development. Improving information flows between software development and operations, therefore, is a sound goal for the entire defense industry.


#selection at the end -add back the deselected mirror
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the

That the DoD struggles with delivering software to the forces in a timely manner is well documented (Brady, 2020; Pomerlau, 2016). This has prompted the DoD to look for new ways of speeding up the development and delivery of software-intensive combat systems. As part of these efforts, the DoD has sought to adopt best practices from private industry and Silicon Valley, in particular (Freedberg, 2020). These practices include agile software development and DevOps, which is the integration of software development (the "Dev") and operations (the "Ops") (Kramer & Wagner, 2019). The DevOps concept seeks to bring developers and operators into a harmonious relationship to improve communication, increase development speed, and reduce the rate of errors and inefficiencies in the implementation of new technology.

> **DevOps is not just a new way of doing things, but DevOps is a new way of thinking about how the Navy develops and delivers combat systems.**

Adopting new ways of work such as DevOps is never an easy task for large organizations such as the Navy. This article investigates the following question: *What are the challenges of adopting DevOps in the U.S. Navy for combat systems?* To identify the obstacles, we start with a literature review of DevOps implementation in industry. We use the literature review to develop interview instruments and conduct 11 semistructured interviews with subject matter experts in defense software acquisition. The article categorizes the interviewees' observations according to the types of challenges cited by the interviewees. For each challenge, we elaborate on the issue facing the Navy and how industry has approached the issue.

Background on DevOps and Agile Methods

DevOps, as illustrated in Figure 1, creates a work environment where development, testing, and operations are part of a single infinite cycle (Kim et al., 2016). The DevOps concept is a large departure from traditional DoD systems engineering models such as the Vee model that assumes top-down and sequential development with a specific delivery date and transition into operations. In DevOps, the developers provide a product to the testers who certify that it is safe, reliable, and capable of meeting an operational need. The testers pass testing results back to the developers in near real-time so the data can be used to make improvements and to fix shortfalls within the software or hardware. With these improvements made, the new change or fix is deployed to the fleet. In the case of the Navy, the operators and users are sailors stationed on ships or submarines who both operate and maintain combat systems. After using the combat systems, sailors and their commanders then provide feedback to the developers and testers so that they can improve the combat system and the testing methods used to certify it for operation. Unlike current system development processes in the Navy, this cycle happens for the duration of the life of the program.

FIGURE 1. TYPICAL DEVOPS CYCLE

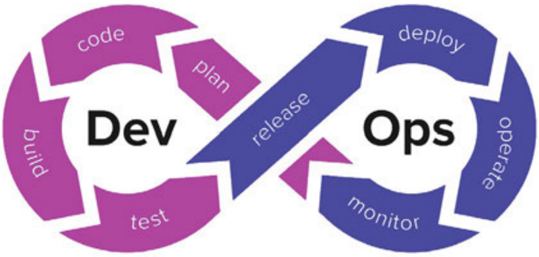


TABLE 1. LIST OF DEVOPS CONCEPTS

DevOps Concept
Open Communication and Close Collaboration
Continuous Experimentation
Continuous Feedback
Continuous Integration
Operational Flow

While we describe DevOps as a process, it is enabled by a set of philosophies and practices intended to increase the speed of development and delivery of capabilities while still ensuring the efficacy and safety of those capabilities. DevOps is not just a new way of doing things, but DevOps is a new way of thinking about how the Navy develops and delivers combat systems. DevOps requires a tighter integration of the development, testing, and operations of these combat systems into a symbiotic web of constant improvement. DevOps delivers value when organizations adopt the key concepts listed in Table 1.

Literature Review

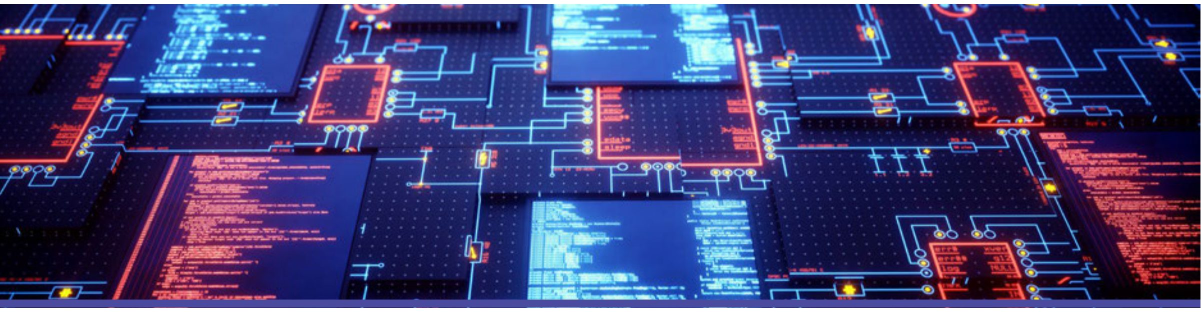
The adoption of DevOps by organizations has been widely studied in the software industry, especially for organizations delivering software as a service or purely software products. The majority of the work is based on case studies. Table 2 summarizes a few of the studies on challenges of DevOps' adoption. In the government sector, there are far fewer papers. Cagle et al. (2018) describe how federal organizations continue to struggle with adoption of agile processes, and they recommend changes to requests for proposals that can help because contractors perform much of the software development. Morales et al. (2018) recommend questions to consider when implementing DevOps in highly regulated environments like the DoD and suggest only a partial DevOps might be possible. Robertson and Bonner (2020) state how agile software practices were developed to capitalize on particular customer characteristics in the commercial sector, and for the DoD to be successful with DevOps, it must tailor the agile concepts and practices to its unique situation.

TABLE 2. CHALLENGES OF DEVOPS ADOPTION CITED IN LITERATURE

Source	Challenges Cited	Citation
Senapathi et al. (2018)	Resistance to cultural change and work process change Staff skills recruitment Tools adoption	6 interviews in a single company
Luz et al. (2019)	Automation Transparency and data sharing Continuous measurement Quality assurance	15 interviews in a single company
Lwakatare et al. (2016)	Culture of continuous improvement Test management Deployment process automation Feedback of operational data	Case study and interviews of 4 European companies developing embedded systems
Riungu-Kalliosaari et al. (2016)	Communication between operations and development Culture changes to implement DevOps Industry constraints on data sharing	Case study and interviews of 3 European companies
Leite et al. (2019)	Process redesign for continuous delivery Tool integration	Literature review of DevOps papers

The use of DevOps for combat systems involves the tight integration of hardware and software, which is called embedded systems in the literature. Chaillan and Yasar (2019) note DevOps remains a problem for embedded and real-time systems in the DoD, which includes combat systems. A 2016 paper claimed there was no evidence of DevOps in the embedded domain (Lwakatare et al., 2016).

The review of the literature shows most studies have examined consumer software and software as a product. Far fewer papers address DevOps challenges in the DoD—a highly regulated environment—nor is there much experience on DevOps for embedded software as found in combat systems. This article contributes to the literature a study of the obstacles to adoption of DevOps in the U.S. DoD.



Research Method

Our research question was *What are the challenges in implementing DevOps in the Navy?* To address the question, we used a qualitative research method of semistructured interviews. Qualitative research provides a rich and effective means to identify the factors or issues affecting one or more outcomes (Kvale, 1994). In our case, we use the qualitative research approach to identify those factors obstructing adoption of DevOps. The research goal is to classify and describe the challenges unique to Navy acquisition in adopting DevOps.

To research the topic of DevOps adoption, we started with a literature review of DevOps and its implementation in industry (see Miller [2020] for full literature review). We relied heavily on the change management literature and viewed the research question through the lens of change management theories. We categorized challenges to DevOps adoption, and used this information to develop and organize our interview questions. The semistructured interviews lasted anywhere from 45 minutes to 2 hours and consisted of five to six starting questions concerning the technical, cultural, regulatory, and process challenges the Navy must confront in its attempt to adopt best practices for software development. The interviewers asked the respondents to draw upon their professional experiences, current work in the field, and knowledge of both the Navy's acquisition programs as well as those in private industry. Table 3 lists the subject matter experts who were interviewed, along with their positions and brief descriptions of their expertise.

TABLE 3. INTERVIEWEES

Interviewee	DevOps Experience
Department of Navy (DoN) contractor for Naval Air Systems Command	Over 30 years as a Naval Officer and 13+ years working as a contractor and Agile consultant.
Senior Software Engineer for Naval Information Warfare	Over 25 years working with Navy IT systems and championing DevOps and Agile practices.
Chief Engineer at DoD contractor	Over 20 years as a Naval Officer and 15+ years as a software developer implementing Agile and DevOps practices.
Project Manager at Program Executive Officer (PEO) Integrated Weapons Systems	Over 15 years as an officer in the Naval Reserve and is employed doing IT development using DevOps and Agile practices in civilian life.
DoN contractor at PEO Integrated Weapons Systems	Over 40 years as a Naval Officer and civilian Acquisition Professional with a focus on combat systems certification and testing.
Program Manager at Naval Air Systems Command	Over 20 years as a Naval Officer and Acquisition Professional with a background in rapid prototyping and Agile development.
Senior Software Engineer at PEO Integrated Weapons Systems	Over 15 years developing and testing Naval weapons and cyber systems.
Assistant Program Manager at PEO Integrated Weapons Systems	Over 10 years as a Naval Officer and Acquisition Professional.
Assistant Program Manager at PEO Soldier	Over 15 years of enlisted and commissioned experience in the Army and a background in Agile development.
Scrum Master at Air Force's Kessel Run Program Office	Over 10 years' enlisted, commissioned in the USAF, and formal education in IT systems.
Systems Certification Manager at PEO Integrated Weapons Systems	Over 10 years as a DoN civilian.

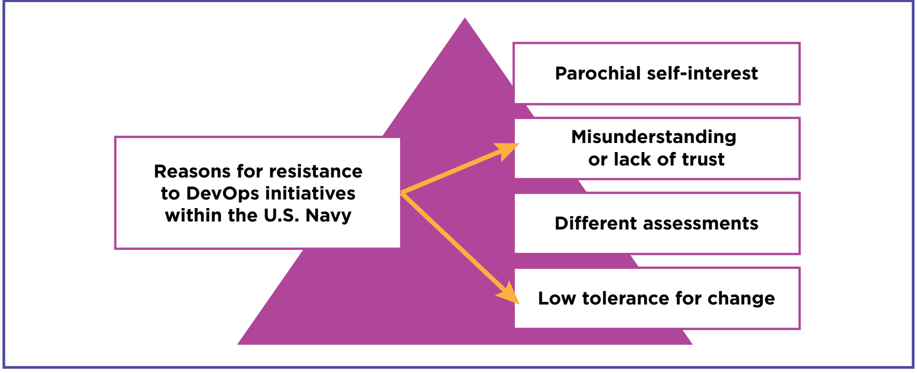
Interview Analysis

This section describes the analysis of the interview observations. We classified the interview observations according to the type of challenge, and we link it to the relevant literature. When appropriate, we quote and paraphrase the interviewees.

Cultural Challenges

Resistance to the introduction of new ways of working is common, and DevOps involves significant changes to work flow, job description, and other aspects of the work environment. Gibson et al. (2012) identified four key reasons for resistance to change: (a) the self-interest of the resister, (b) the resister's misunderstanding of the change, (c) the resister having a different assessment of the best course of action for change, and (d) a low tolerance within the organization for change. Within the Navy, respondents identified the second and fourth reasons as the most pressing (see Figure 2).

FIGURE 2. PRIMARY REASONS FOR RESISTING CHANGE TO DEVOPS WITHIN U.S. NAVY


Reasons for resistance to DevOps initiatives within the U.S. Navy
Parochial self-interest
Misunderstanding or lack of trust
Different assessments
Low tolerance for change

Note. Adapted from Gibson et al. (2012).

The Navy has a history of resistance to change due to simple institutional inertia (Hall & O'Connor, 2018) as well as a desire to preserve tradition (Buhl, 1974). As was noted by one of the respondents who served in the Navy, a sailor's favorite phrase when asked why a task is completed a certain way is "That's the way we've always done it" (Anonymous acquisition professional personal communication, February 10, 2020). This deep-rooted resistance to change can make implementing DevOps a difficult task. The Air Force found the hardest hurdle to overcome when implementing DevOps practices within the F-22 Raptor program was to change the program's culture (Ulsh & McCarty, 2019).

One cybersecurity engineer stated that the misunderstandings about the nature of DevOps and resistance to change within the organization have hindered prior attempts to adopt DevOps and agile business practices. Once again, this is reflected across the entire DoD. In an annual survey of major acquisition programs in 2019, the U.S. Government Accountability Office found that of the 22 programs that claimed to be agile, only six conformed to the best practices of private industry (Freedberg, 2020).

Risk Aversion

Chief among the cultural hurdles is the Navy's aversion to risk in acquisition programs. One respondent stated that the Navy's budget has decreased since the Cold War, yet combat systems costs continue to increase, which creates an environment where acquisition programs are increasingly wary of any and all risks. This has resulted in a climate where, as one program manager at Naval Air Systems Command (NAVAIR) lamented, there is, "no tolerance for risk within the Navy acquisitions and development hierarchy" (Anonymous acquisition professional, personal communication, February 10, 2020). This makes any deviation from established norms difficult to implement, and this sentiment is contrary to the DevOps culture of an environment where personnel feel it is safe to take risks and potentially fail (Forsgren et al., 2018). DevOps adheres to the fail-fast mentality because it encourages developers to test ideas, emphasizes the value of the knowledge gained by any failures, and allows developers to be more creative in responding to emergent system requirements.

Aversion to risk manifests itself in the regulatory environment in which the Navy operates. Of the 11 respondents, eight said the Navy's attempts to innovate are stifled by the rigid statutes and regulations required by Congress to ensure the government shoulders as little risk as possible in acquiring new combat systems. These statutes and regulations are written into DoD and Navy policies that dictate how to write contracts, how to decide upon contract awards, and how the government's money can be spent. As one consultant at Naval Sea Systems Command (NAVSEA) opined, "The reason we have the rules we have is because we messed up in the past, and needed to codify rules to prevent those screw-ups in the future" (Anonymous acquisition professional, personal communication, February 10, 2020).

> **One respondent stated that the Navy's budget has decreased since the Cold War, yet combat systems costs continue to increase, which creates an environment where acquisition programs are increasingly wary of any and all risks.**

During the interview process, every respondent expressed disfavor with the way contracts were written and awarded. One acquisition professional at NAVSEA stated, "We make contract awards based upon price alone" (Anonymous program manager, personal communication, February 10, 2020). The manager for a program in the Air Force mentioned, "Military contracts make things too specific" and because of that "We can't provide the best solution" to the warfighter (Anonymous program lead, personal communication, February 14, 2020).

The Navy's need to specify all requirements in contracts is problematic for DevOps (Jacobs & Kaim, 2021c). In DevOps, the developers work closely with operators to determine their needs and identify the work for the next sprint during the Scrum process (Forsgren et al., 2018; Gilman et al., 2019). During these sprints, developers and operator advocates meet daily to ensure that the work being performed aligns with the needs of the customer regarding functionality and user experience. This close working relationship builds trust between the customer and the developer, and nullifies the need for rigid contract language to ensure that the developer will deliver an acceptable product. To adopt DevOps, the Navy will need to overcome these trust barriers and build closer working relationships with its contractors.

Vendor Lock-in

Risk aversion affects not only the Navy, but also the contractors who design and build the Navy's systems. Because development expenses are so high and profit margins so thin, contractors meticulously protect their intellectual property (Gilman et al., 2019). As one contractor mentioned, the result is the Navy relies on proprietary systems and the original contractor is the only one qualified to perform follow-on integration work and capability updates (Anonymous agile consultant, personal communication, February 18, 2020). Because of the vendor lock-in effect, it is very important to contractors to win the initial award and, as a result, they tend to take a conservative approach and contest contract awards when they lose.

> **If the Navy continues to rely on proprietary software and not open standards, then it will continue to be unable to keep up with the pace of change in both the private sector and its adversaries.**

Contrast this with private industry that uses open source tools and software (Anonymous senior scientist, personal communication, February 18, 2020). By relying on open source solutions, private industry is able to leverage a larger pool of vendors and contractors, and therefore more possible ideas for solutions. As one consultant working on unmanned aerial systems remarked, "We need to be better about designing open systems" so that "we're not tied to proprietary software or hardware" (Anonymous agile consultant, personal communication, February 18, 2020). Failure to adopt open source solutions will cause continual problems within systems development

programs. For instance, USS *Zumwalt* (DDG 1000) was initially designed to use computer servers from Microsoft, but during the middle of development for Zumwalt, Microsoft sold off its server hardware division. Because the requirements for the ship were written specifically for Microsoft hardware, this change led to unnecessary delays due to the slow requirements generation and approval process. If the Navy continues to rely on proprietary software and not open standards, then it will continue to be unable to keep up with the pace of change in both the private sector and its adversaries.

High Reliability Organization

One interviewee said, "The Navy has to live in a world where we kill people and break things"; consequently, there is little room for bugs or defects in the systems that are given to the fleet (Anonymous cyber engineer, personal communication, February 6, 2020). The Navy has the organizational mindset of a high-reliability organization (HRO). An HRO operates specialized systems, which are deeply interconnected, are potentially hazardous, and have a high risk of catastrophic failure (Shrivastava et al., 2009). The combat systems the Navy develops are complex and tightly coupled, meaning that errors are difficult to diagnose and any defects can potentially propagate quickly throughout the system of systems (Roberts, 1990; Van Stralen, 2017). Example HROs are found in nuclear power plants and airlines.

Many software organizations such as Microsoft or Google do not need the same level of quality assurance as HROs. If Google or Microsoft push a software update that breaks their users' systems, they simply have to launch a media campaign to apologize. If the Navy accepts a defective combat system, then sailors and civilians can possibly die. In the context of Navy DevOps, the culture of high reliability results in approvals for release being slower to achieve, testing being more thorough, and requirements for quality control being more stringent. This will inevitably mean that the Navy cannot achieve the same level of development speed as the technology industry leaders it seeks to emulate since it has higher standards to meet. But, as one acquisition professional at NAVAIR expressed, “DevOps may not make us too much faster, but it’s going to make us on time” (Anonymous agile consultant, personal communication, February 18, 2020).

As one program manager mentioned, the change leader must “defeat the antibodies to change within the Navy’s bureaucracy” (Anonymous program manager, personal communication, February 14, 2020). Another software engineer mentioned, “The novelty of the idea of DevOps and the confusion surrounding just what exactly it is, has resulted in many in the Navy’s upper echelons of leadership not understanding what must be done to bring about change or how to communicate its necessity” (Anonymous cyber engineer, personal communication, February 6, 2020). An agile consultant stated that this is further compounded by the short duration in which leaders remain in command (typically 2 to 3 years) and their high turnover rate (Anonymous agile consultant, personal communication, February 10, 2020). This makes it difficult to carry out long-term change leadership, especially for something like the adoption of DevOps that will likely take a decade or more to be fully realized. A consultant at NAVAIR lamented, “The Navy needs someone at the [Senior Executive Service] or Flag level to lead the charge.” That same consultant also stated, “The Navy needs a character like Hyman Rickover with passion, drive, and horsepower to get the organization charged and aligned to the future of DevOps” (Anonymous agile consultant, personal communication, February 10, 2020).

Regulatory and Process Challenges

All the respondents were quick to point out that the Navy and the DoD labor under many regulations and statutory requirements dictating how they will operate, acquire new combat systems, and perform development of new technologies. Navy acquisition exists in what is known as a highly regulated environment (HRE). An HRE is an environment in which heightened security, access controls, segregation of duties, inability of personnel to discuss certain topics outside of specific areas, and the inability to take certain artifacts off premises are put in place (Morales et al., 2018). An HRE is used when the intellectual property and methods being developed must be safeguarded from theft and all parties involved are sworn to secrecy. This directly conflicts with the DevOps tenet of sharing information openly and freely between all parties involved with the development of a system (Kim et al., 2016).

> **The combat systems the Navy develops are complex and tightly coupled, meaning that errors are difficult to diagnose and any defects can potentially propagate quickly throughout the system of systems.**

The Navy has, as one contractor for an ACAT I program mentioned, "a certain level of stricture and structure that makes it harder to implement DevOps than the civilian sector" (Anonymous chief engineer, personal communication, February 18, 2020). The Navy must limit the open sharing of capabilities, limitations, and technical details about its combat systems because release of such information is a security threat. This forces the Navy to work around "security concerns, classified information, non-ideal hardware restrictions," as well as compartmentalization of vendors (Gilman et al., 2019). But the requirements for secrecy are not the only regulations that the Navy must abide by when developing combat systems. The Federal Government also imposes strict requirements on the funding (Critical Cost Growth, 2012), acquisition strategy (Acquisition Strategy, 2015), and testing and evaluation of new systems (Deputy Assistant Secretary of Defense, 2011; Operational Test & Evaluation, 2010). This means that any new system must meet certain milestones and performance criteria before being accepted, and that failure to do so may end in the program being canceled (Gilman et al., 2019). This contradicts the best practices of DevOps that dictate that the capabilities of a system should be built gradually. To place this in colloquial terms, DevOps requires that an elephant be eaten a bite at a time with small, frequent updates (Senapathi et al., 2018), whereas the DoD acquisition process requires the elephant be eaten all at once.

Evolving Requirements

The Joint Capabilities Integration and Development System (JCIDS) process determines the requirements for a system and involves the generation of many documents including the Initial Capabilities Document (ICD), the Capabilities Determination Document (CDD), and other documents, which are reviewed and need approval during milestones A and B (Chairman of the Joint Chiefs of Staff, 2018). JCIDS is a top-down and plan-based approach, generating a stringent documentation of requirements in a legal manner suitable for a contract (Manning, 2020). The requirements process is the initiation of any acquisition program and forms the basis for all design and engineering decisions that will be made within that program.

> **All the respondents were quick to point out that the Navy and the DoD labor under many regulations and statutory requirements dictating how they will operate, acquire new combat systems, and perform development of new technologies.**

The requirements documents should incorporate feedback and input from sailors and officers in the fleet. Unfortunately, that isn't the case as an acquisition consultant at the Navy's Program Executive Office Integrated Warfare Systems relayed, "Currently Sailors have little say in what goes into combat systems" (Anonymous senior scientist, personal communication, February 18, 2020). Furthermore, the requirements in the CDD are often written to describe specific functions, instead of outcomes for the fleet, reflecting "what the Navy needs" (Anonymous senior scientist, personal communication, February 18, 2020; Thompson, 2019). The plan-based approach is in opposition to how requirements are defined in a DevOps system that puts customer needs as the top priority (Kim et al., 2016). These needs are captured during daily or weekly scrum meetings in which the sprints (development periods) are planned. In each of these scrum meetings, a customer advocate champions the needs of the customer to ensure that the finished products are satisfactory (Barrett & Claxton, 2005). The guiding principle in private industry is to provide value to the customer and focus on outcomes for them (Anonymous testing manager, personal communication, February 10, 2020). This means that if the finished product is functional but doesn't provide exactly what the customer is looking for, or if the user experience is subpar, then it is considered a failure. The possibility of failure is avoided by meeting frequently with the customer advocate to review the progress being made and determine whether what is being developed still meets their needs or not. By implementing direct feedback from the customer, the developers are better able to provide successful products.

Certification and Testing Process

Testing is an integral part of the DevOps process. But unlike private industry, where all testing is rolled into one constant cycle, the Navy conducts testing in discrete stages that are tied directly to milestones in the combat system's life cycle. Furthermore, the Navy differentiates between developmental test and evaluation (DT&E) and operational test and evaluation (OT&E) (Barrett & Claxton, 2005). DT&E, performed during the technology development and engineering and manufacturing phases of development, is conducted to prove design concepts, demonstrate technological maturity, and identify integration problems prior to final prototyping. OT&E, carried out at the end of the engineering and manufacturing development phase, uses the actual system to determine whether the system is operationally effective and operationally suitable (Anonymous acquisition professional, personal communication, February 12, 2020).

Testing within the Navy is carried out under the authority of the program management staff with either the contractor (in the case of DT&E) or the Navy's operational testing command (in the case of OT&E) performing the testing. This testing must be conducted within the guidelines of the DoD's Director of Developmental Test and Evaluation and Director of Operational Test and Evaluation, respectively (Ullman, 2019). All testing is based on, and conducted in accordance with, each program's test and evaluation master plan (TEMP), which derives from the capabilities documents produced during the initial design phase of development via the DoD requirements generation process. This TEMP ties testing events to specific capability requirements as well as to development milestones and serves as the "contract between program management staff, systems integration experts, the contractors, and [Navy's operational testing command] for what is to be tested," as one contractor at NAVSEA explained (Kramer & Wagner, 2019).

This testing procedure was born out of a need for the Navy to develop complex hardware systems and to prove their efficacy prior to delivering them to the fleet. This need to develop hardware alongside the software forces the developers to design tests and divert resources for the test equipment necessary to perform those tests for the hardware (Ullman, 2019). This means that any development of combat systems using DevOps must include detailed test planning and plentiful developmental testing early in the project (Anonymous acquisition professional, personal communication, February 12, 2020). Hardware development also results in rigid testing schedules that do not respond well to changes or delays. As one expert in how the Navy performs testing revealed, the TEMP usually takes years to get approved, as it has to be reviewed by program management staff, systems integration experts, the Navy's Warfighting Requirements and Capabilities Office (which is responsible for system requirements and resource allocation), and the DoD directors for test and evaluation (Jacobs & Kaim, 2021a). This expert's example was the TEMP for USS *Gerald R. Ford* (CVN 78). The *Ford's* TEMP took 10 years to make it "through the labyrinth of bureaucratic red tape" for approval (Anonymous acquisition professional, personal communication, February 12, 2020) because every time a change was made in the technology being used during the ship's decades of development, the TEMP had to be updated and go back through the entire approval process from the start. This delayed testing and ultimately the final delivery of the ship to the Navy.

Contrast this with the way testing is performed in a DevOps environment where the prevailing theory is to break the software early and often so that weak points and inefficiencies in the code can be discovered and fixed quickly (Hofmann et al., 2018). Using these agile testing practices, a system can be updated and improved rapidly due to the massive amount of data available to the developers to identify problems and adjust code or hardware components. Once again, the goal for testing in a DevOps environment is to shorten the time it takes to build a system, test it, and put the results from those tests back in the hands of developers (Shahin et al., 2017). Like the Navy, strategic planning of testing is needed to ensure that the testing is adequate for pushing the system to its limits and testing its functionality. Oftentimes, this planning is performed using software that integrates directly with the testing suite to provide better collaboration throughout different departments in the company (Gilman et al., 2019). Also like the Navy, private industry leverages cross-team testing where the team responsible for testing is different than the team that developed the product. It is claimed that this cross-pollination of testing and development teams allows for the detection of defects faster. The Navy must adapt its current testing practices to provide for better cross-team collaboration and a higher volume of tests in a shorter amount of time.

Software Certification and Testing Process

Software systems have a second analogous process that they must undergo to be approved for use on Navy computer networks. This process is part of the DoD Information Assurance certification process and results in the software earning an authority-to-operate certification (Anonymous senior scientist, personal communication, February 18, 2020). The main focus of this certification process is to ensure the security and integrity of the DoD's IT systems. Similar to the systems certification process, software must be tested against security requirements; those test results must then be reviewed by an authorizing official, and upon successful completion, that official issues the certification allowing the software to be loaded onto Navy computers and servers.

> **The goal for testing in a DevOps environment is to shorten the time it takes to build a system, test it, and put the results from those tests back in the hands of developers.**

Like the systems certification process, the Information Assurance certification process moves at a slow pace and requires manual approvals and initiation of testing at specific program milestones (Obicci, 2017). Every expert interviewed mentioned that the certification process moves too slowly and the requirements needed to achieve certification are too cumbersome. A computer scientist at Naval Information Warfare Systems Command (NAVWAR) mentioned that the ideal process would allow for a continuous certification as well as cross-compatibility between systems so that software is able to be loaded on any system once it is deemed "safe."

Now contrast the slow pace of the Information Assurance certification process with the continuous Risk Management Framework (RMF) process that is typically used in private businesses (Obicci, 2017). The RMF applies the same ideas of continuous learning and integration to information security to reduce the time it takes to detect security threats and respond to them (Anonymous cyber engineer, February 6, 2020). The RMF does this in a simple process of identifying potential risks, prioritizing those risks based upon their threat to the customer and developer, developing mitigation strategies, enacting those strategies, and then testing them (Ullman, 2019). These test data are then fed back into threat assessment. This means that the threat assessment is a continuous, ongoing process instead of a single event.

> **Similar to the systems certification process, software must be tested against security requirements; those test results must then be reviewed by an authorizing official, and upon successful completion, that official issues the certification allowing the software to be loaded onto Navy computers and servers.**

Technical Challenges

Though the nontechnical impediments took center stage during the interviews, there were still two technical challenges that the respondents were not sure how to address. These challenges were how to perform hardware development using DevOps or agile methods, and how to implement data feedback loops while still fulfilling the requirements for data security and classification. The three engineers who were questioned were confident that technologies and software used in private industry could be adapted for purposes of the Navy's acquisition programs. As one mentioned, "Private industry has been doing [DevOps and agile] for years" and the software is "out there" and readily available (Jacobs & Kaim, 2021b).

Hardware DevOps

The largest technical challenge preventing the Navy from implementing DevOps is that the Navy's combat systems comprise both hardware and software. Software development is iterative and incremental, with each update and patch serving as a building block towards the overall capability of the system (Schuh et al., 2016a; Ullman, 2019). Contrast the iterative and incremental approach with hardware programs that view the addition of capability requirements as requirements "creep." This creep often causes ambiguity in the primary capabilities of the system, difficulties in systems integration, subpar systems performance, cost overruns, and eventually project cancellation (Schuh et al., 2016b). The defense acquisition process is intentionally rigid because they want to establish requirements early in the design process and avoid costly design revisions and physical rework during manufacturing.

To date, very little literature has been written regarding the adaptation of DevOps or agile methodologies to hardware development. This is due to the nature of hardware development and the need to invest heavily in up-front material costs for hardware as well as in testing equipment (Anonymous acquisition professional, personal communication, February 10, 2020). The use of DevOps practices within hardware development is also confounded by the need to develop hardware on which to run software (Anonymous senior scientist, personal communication, February 18, 2020). Unlike pure software systems and development programs, many of the systems like radars and missiles are not hardware-agnostic and need specific hardware developed to meet operational needs. This means that the hardware must be developed before or in conjunction with the software. This forces the need to find ways in which to divorce the development and updating of software from hardware (Anonymous program manager, personal communication, February 10, 2020).

Separating the software from the hardware would make for simpler development programs and allow software development to be unhindered by hardware limitations. As one scientist at NAVWAR explained, "The hardware update tempo is much slower. Software can be updated daily but hardware takes years" (Anonymous senior scientist, personal communication, February 18, 2020). That being said, due to the Navy's culture as an HRO, this means that any integration of hardware and software must still be able to function safely and reliably. Furthermore, hardware becomes obsolete much faster than software. In the current "waterfall" Defense Acquisition System that takes a decade or more to come to fruition, this obsolescence of hardware creates a "tech refresh spiral" that leads to nearly "endless requirements creep and the eventual death of programs" (Anonymous acquisition professional, personal communication, February 10, 2020). These facts mandate that any adoption of DevOps methods in the hardware domain make sufficient use of configuration management tools to ensure functional integration of differing levels of hardware maturity. As the same scientist at NAVWAR clarified, "The goal isn't how to do agile hardware but how to manage obsolescence" (Anonymous project manager, personal communication, February 12, 2020).

Data Feedback

DevOps depends on the ability of the organization to collect and distribute continuous feedback on the combat systems to the developers. Gathering operational data on a system is not an entirely foreign concept to the Navy because data are required for the Defense Information Assurance certification and systems certification processes. However, data collection in the Navy is neither automated nor continuous. A testing manager at NAVSEA said, "The Navy currently relies upon instrumentation for tests that must be installed manually and combat systems must be configured to collect and store detailed data" (Anonymous senior scientist, personal communication, February 18, 2020). The data must then be manually packaged and couriered back to developers and engineers for analysis, as there is no automatic system to collect and transmit the data back ashore to developers (Anonymous assistant program manager, personal communication, February 17, 2020).

The Navy not only lacks the infrastructure to automatically collect and distribute the data, but it also lacks the personnel needed to make sense of all the data. A software engineer at NAVWAR explained data analysts in private industry are often used to analyze and interpret data to answer questions such as, "Are we effective?" or "Can we accomplish the mission better?" (Anonymous acquisition professional, personal communication, February 18, 2020). These data analysts play a crucial role in finding connections between the data and root causes of subpar performance. They can also play a role in better understanding customer needs. For instance, when a customer says that a user interface is "bad," the data analysts can perform analysis and find data to show that what the customer meant by "bad" was actually slow loading times.

> **The sharing of data between operational and developmental organizations as well as between government and contractors goes against the normal way the Navy does business.**

This kind of interpretation for the customer is no less important in the DoD. As one assistant program manager at PEO Soldier explained, the Army needs the data feedback and analysis to understand "how better physical training scores correlate with better marksmanship" (Meyer, 2014). Such feedback can help the Services better design the systems including the nonmateriel aspects of doctrine, training, and so forth encompassed by the acronym DOTmLPF-P (Doctrine, Organization, Training, Materiel, Leadership, Personnel, and Facilities-Policy). Unfortunately, the Navy and the DoD as a whole lack the number of data analysts needed to support all of their acquisition programs (Meyer, 2014). This is a critical need that must be filled for DevOps to work.

Increased Exposure to Security Risks

Within the DoD, the more common term is actually DevSecOps to emphasize the importance of ensuring security because the continuous updating and feedback leads to greater exposure to security risks. The sharing of data between operational and developmental organizations as well as between government and contractors goes against the normal way the Navy does business.

Conclusions

Throughout the 11 interviews and through all of the correspondence gathered from respondents, a trend became clear that despite the technical nature of DevOps, the respondents' largest concerns were with the cultural, organizational, process, and regulatory hurdles that stand in the way of the Navy adopting DevOps. The adoption of DevOps requires a drastic organizational and cultural shift within the Navy to establish the necessary work processes, individual training, responsibility, and policies. Table 4 shows the challenges identified through analysis of the interviews and classifies them according to the DevOps concepts identified in the literature review. During the interview process, it became apparent that these challenges are all interconnected and will require a holistic approach to change. The respondents all said the nontechnical problems must be addressed before any technical solution can be found. The two main technical challenges were (1) how to perform DevOps for hardware and (2) how to establish an infrastructure for collecting and using feedback data from the fleet to design and build better combat systems.

TABLE 4. DEVOPS CONCEPTS AND ASSOCIATED CHALLENGES

DevOps Concept	Challenges
Open Communication and Close Collaboration	• Rigid organizational hierarchy • Security requirements • Cultural inertia
Continuous Experimentation	• Statutory requirements • Rigid test processes • Rigid requirements generation process
Continuous Feedback	• Lack of infrastructure • Security requirements • Cultural inertia
Continuous Integration	• Cultural inertia • Rigid test processes • Rigid requirements generation process • Hardware requirements
Operational Flow	• Entrenched cultural practices • Rigid work processes • Rigid requirements generation process • Hardware requirements

The research used semistructured interviews to collect data from 11 SMEs. A limitation of the research is the small sample size compared to the large size and diversity of software development across program offices in the Navy and DoD. However, the findings agree with the challenges identified by the literature for commercial software systems (see Table 2). Our interviewees emphasized the security concerns and that cultural and organizational changes were necessary, although difficult to address in the DoD because of the regulations and entrenched culture. Knowing that the challenges resemble those found in commercial industry is useful because it suggests that the Navy can adapt and apply many of the industry approaches to overcome them. Understanding the obstacles facing adoption of DevOps is important for theoretical and practical reasons. First, this knowledge can help researchers bridge disconnected insights at the national and individual levels. Second, this knowledge can also help acquisition leaders develop plans and prepare interventions to support adoption of DevOps.

References

Acquisition Strategy, 10 U.S.C. § 2431 (2015). https://www.govinfo.gov/app/details/USCODE-2015-title10/USCODE-2015-title10-subtitleA-partIV-chap144-sec2431a

Barrett, C., & Claxton, J. D. (Ed.). (2005). *Test and evaluation management guide* (5th ed.). Defense Acquisition University. https://www.dau.edu/tools/t/Test-and-Evaluation-Management-Guide-(TEMG)

Brady, S. (2020, November 11–13). DoD's software acquisition pathway digital delivery at the speed of relevance. National Defense Industrial Association Virtual Systems and Mission Engineering Conference, United States. https://www.ndia.org/events/2020/11/10/2020-vsystems-and-mission-engineering-conference/speakers

Buhl, L. C. (1974). Mariners and machines: Resistance to technological change in the American Navy, 1865–1869. *Journal of American History, 61*(3), 703–727. https://doi.org/10.2307/1899928

Cagle, R., Kristan, M. J., & Rice, T. (2018). *DevOps for federal acquisition*. MITRE. https://www.mitre.org/publications/technical-papers/devops-for-federal-acquisition

Chaillan, N., & Yasar, H. (2019). *Waterfall to DevSecOps in DoD*. Carnegie Mellon University Software Engineering Institute. https://apps.dtic.mil/sti/pdfs/AD1085204.pdf

Chairman of the Joint Chiefs of Staff. (2018). *Charter of the Joint Requirements Oversight Council (JROC) and Implementation of the Joint Capabilities Integration and Development System (JCIDS)* (CJCSI 5123.01H). https://www.jcs.mil/Portals/36/Documents/Library/Instructions/CJCSI%205123.01H.pdf?ver=2018-10-26-163922-137

Critical Cost Growth in Major Defense Acquisition Programs, 10 U.S.C. § 2433a (2012). https://www.govinfo.gov/app/details/USCODE-2011-title10/USCODE-2011-title10-subtitleA-partIV-chap144-sec2433a

Deputy Assistant Secretary of Defense for Developmental Test and Evaluation; Deputy Assistant Secretary of Defense for Systems Engineering: Joint Guidance, 10 U.S.C. §§ 139b, d (2011). https://www.govinfo.gov/app/details/USCODE-2010-title10/USCODE-2010-title10-subtitleA-partI-chap4-sec139b

Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science of lean software and DevOps: Building and scaling high performing technology organizations*. IT Revolution Press. https://itrevolution.com/book/accelerate/

Fullerton (2003).

Freedberg, S. J., Jr. (2020, June 12). DoD 'agile' software development still too slow: GAO. *Breaking Defense*. https://breakingdefense.com/2020/06/dod-agile-software-development-still-too-slow-gao/

Gibson, J. L., Ivancevich, J. M., Donnelly J. H., Jr., Konopaske, R. (2012). *Organizations: Behavior, structure, processes* (14th ed.). McGraw-Hill Irwin. http://dl.motamem.org/organizations_behavior_structure.pdf

Gilman, L., Medin, M., Pahlka, J., & Stephens, T. (2019). *Software is never done* (1st ed.). Defense Innovation Board. https://doi.org/10.1017/CBO9781107415324.004

Hall, J. S., & O'Connor, J. M. (2018). *Learning technology adoption: Navy barriers and resistance*. Calhoun: NPS Institutional Archive. http://hdl.handle.net/10945/58306

Hofmann, C., Lauber, S., Haefner, B., & Lanza, G. (2018). Development of an agile development method based on Kanban for distributed part-time teams and an introduction framework. *Procedia Manufacturing, 23*, 45–50. https://doi.org/10.1016/j.promfg.2018.03.159

Jacobs, M., & Kaim, E. (2021a). *Shift left to make testing fast and reliable*. Azure DevOps, Microsoft Docs. https://docs.microsoft.com/en-us/azure/devops/learn/devops-at-microsoft/evolving-test-practices-microsoft

Jacobs, M., & Kaim, E. (2021b). *What is agile?* Azure DevOps, Microsoft Docs. https://docs.microsoft.com/en-us/devops/plan/what-is-agile

Jacobs, M., & Kaim, E. (2021c). *What is Scrum?* Azure DevOps, Microsoft Docs. https://docs.microsoft.com/en-us/azure/devops/learn/agile/what-is-scrum

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook: How to create world-class agility, reliability* (2nd ed.). IT Revolution Press. https://itrevolution.com/book/the-devops-handbook/

Kramer, J., & Wagner, T. (2019). Developmental test and requirements: Best practices of successful information systems using agile methods. *Defense Acquisition Research Journal, 26*(2), 128–150. https://doi.org/10.22594/dau.19-819.26.02

Kvale, S. (1994). *Interviews: An introduction to qualitative research interviewing*. Sage Publications. https://psycnet.apa.org/record/1996-97829-000

Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys (CSUR)*, 52(6), 1–35. https://dl.acm.org/doi/abs/10.1145/3359981

Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, 157, 1-35. https://doi.org/10.1016/j.jss.2019.07.083

Lwakatare, L. E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H. H., Bosch, J., & Oivo, M. (2016, January 5–8). Towards DevOps in the embedded systems domain: Why is it so hard? In T. X. Bui & Ralph H. Sprague, Jr. (Co-Chairs), *Proceedings of the IEEE 49th Hawaii International Conference on System Sciences (HICSS)* (pp. 5437–5446), Koloa, HI, United States. https://doi.org/10.1109/HICSS.2016.671

Manning, B. D. (2020). *JCIDS process*. AcqNotes. http://acqnotes.com/acqnote/acquisitions/jcids-overview

McQuade, J. M., Murray, R. M., Louie, G., Medin, M., Pahlka, J., & Stephens, T. (2019). *Software is never done: Refactoring the acquisition code for competitive advantage*. Defense Innovation Board. https://media.defense.gov/2019/Mar/26/2002105909/-1/-1/0/SWAP.REPORT_MAIN.BODY.3.21.19.PDF

Meyer, M. (2014). Continuous integration and its tools. *IEEE Software, 31*(3), 14–16. https://doi.org/10.1109/MS.2014.58

Miller, A. W. (2020). *Integrating DevOps into Navy combat systems development* [Master's thesis, Naval Postgraduate School]. Calhoun: NPS Institutional Archive. http://hdl.handle.net/10945/66688

Morales, J. A., Yasar, H., & Volkmann, A. (2018). Weaving security into DevOps practices in highly regulated environments. *International Journal of Systems and Software Security and Protection, 9*(1), 18–46. https://doi.org/10.4018/ijsssp.2018010102

Obicci, P. (2017). *Risk management framework*. IGI Global. https://doi.org/10.4018/978-1-5225-2503-5.ch007

Operational Test and Evaluation of Defense Acquisition Programs, 10 U.S.C. § 2399 (2010). https://www.govinfo.gov/app/details/USCODE-2010-title10/USCODE-2010-title10-subtitleA-partIV-chap141-sec2399

Pomerlau, M. (2016, July 20). DoD acquisition not broken, just slow. *C4ISRNET*. https://www.c4isrnet.com/c2-comms/2016/07/20/dod-acquisition-not-broken-just-slow/

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016, November 22–24). DevOps adoption benefits and challenges in practice: A case study. In *International Conference on Product-Focused Software Process Improvement* (pp. 590-597), Trondheim, Norway. Springer International. https://www.cs.helsinki.fi/u/jutiihon/publications/RiunguKalliosaari2016DevopsAdoptionBenefits.pdf

Roberts, K. H. (1990). Some characteristics of one type of high reliability organization. *Organization Science, 1*(2), 160–176. https://doi.org/10.1287/orsc.1.2.160

Robertson, J., & Bonner, T. (2020, March-April). Enabling software innovation across the DoD. *Defense Acquisition, 49*(2), 30–35. https://www.dau.edu/library/defense-atl/DATLFiles/Mar-April_2020/DEFACQ%20Mar-Apr%202020.pdf

Schuh, G., Lau, F., Schröder, S., & Wetterney, T. (2016a, September 4–8). Next generation hardware development: The role of technology intelligence to reduce uncertainty in Agile new Product Development. In D. F. Kocaoglu (Ed.), *Proceedings of the IEEE PICMET 2016 Portland International Conference on Management of Engineering and Technology: Technology Management for Social Innovation* (pp. 2573–2582), Honolulu, Hawaii, United States. https://doi.org/10.1109/PICMET.2016.7806808

Schuh, G., Schröder, S., Lau, F., & Wetterney, T. (2016b, September 4–8). Next generation hardware development: Requirements and configuration options for the organization of procurement activities in the context of agile new product development. In D. Kocaoglu (Ed.), *Proceedings of the IEEE 2016 Portland International Conference on Management of Engineering and Technology (PICMET)* (pp. 2583–2591), Honolulu, Hawaii, United States. https://doi.org/10.1109/PICMET.2016.7806809

Senapathi, M., Buchan, J., & Osman, H. (2018, June 28–29). DevOps capabilities, practices, and challenges. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering* (pp. 57–67), Christchurch, New Zealand. https://doi.org/10.1145/3210459.3210465

Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. In *IEEE Access* (Vol. 5, pp. 3909–3943). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ACCESS.2017.2685629

Shrivastava, S., Sonpar, K., & Pazzaglia, F. (2009). Normal accident theory versus high reliability theory: A resolution and call for an open systems view of accidents. *Human Relations, 62*(9), 1357–1390. https://doi.org/10.1177/0018726709339117

Sommerville, I. (2021). *Software engineering* (10th ed.). Pearson. https://www.pearson.com/store/p/software-engineering/P100001451389

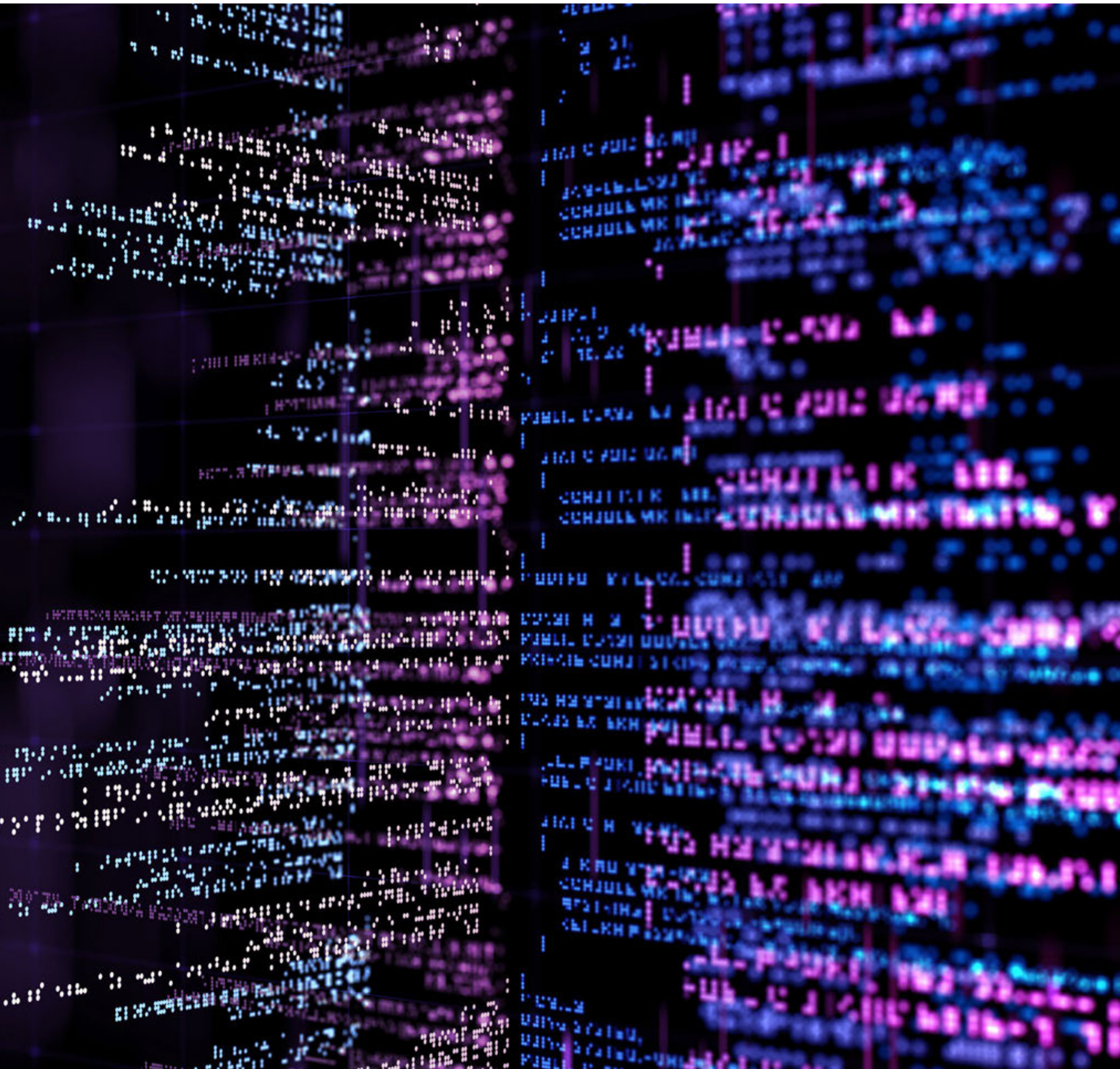
Thompson, E. (2019). *Agile project management: The step by step guide that you must have to learn project management correctly from the beginning to the end* (1st ed.). Amazon Publishing. https://www.amazon.com/Agile-Project-Management-Correctly-Beginning/dp/1072162784/ref=tmm_pap_swatch_0?_encoding=UTF8&qid=&sr

Ullman, D. G. (2019). *Scrum for hardware design* (1st ed.). The Mechanical Design Process. https://www.mechdesignprocess.com/scrum-hardware-design

Ulsh, C., & McCarty, M. Z. (2019). Vignette 2 – F22 : DevOps on a hardware platform. In J. M. McQuade & R. M. Murray (Eds.), *Software Is Never Done* (1st ed., pp. 53–54). Defense Innovation Board. https://innovation.defense.gov/software/

Van Stralen, D. (2017). *HRO models*. High Reliability Organizing. http://high-reliability.org/hro-models

The views expressed in this article are those of the author(s) alone and not of the Department of Defense. Reproduction or reposting of articles from *Defense Acquisition Research Journal* should credit the author(s) and the journal.



Author Biographies

LT Andrew W. Miller, USN

is an Engineering Duty Officer assigned to Naval Information Warfare Center Atlantic in Charleston, South Carolina. His work specializes in the development and acquisition of shipboard and expeditionary computer, software, and communications systems. LT Miller holds a BS in Mechanical Engineering from the Virginia Military Institute and an MS in Systems Engineering from the Naval Postgraduate School.

(E-mail address: andrew.miller@nps.edu)

Dr. Ronald E. Giachetti

is a Professor of Systems Engineering at the Naval Postgraduate School. He teaches and conducts research in systems modeling and architecture. He holds a BS in Mechanical Engineering from Rensselaer Polytechnic Institute, an MS in Manufacturing Engineering from Polytechnic University, and a PhD in Industrial Engineering from North Carolina State University. Dr. Giachetti has published over 50 technical articles on systems modeling and architecture, including a textbook on the Design of Enterprise Systems. He is a member of the Navy's Systems Engineering Stakeholder's Group and cochair of the Corporate Advisory Board of the International Council on Systems Engineering.

(E-mail address: regiache@nps.edu)

Dr. Douglas L. Van Bossuyt

is an Assistant Professor in the Systems Engineering Department at the Naval Postgraduate School in Monterey, California. His research focuses on the nexus of system architecture, risk and failure analysis, resilient system design, digital twin, model-based systems engineering, and design decision-making support. He teaches courses in combat systems integration, reliability and other "-ilities," and system architecture. Dr. Van Bossuyt holds an Honors Bachelor of Science in Mechanical Engineering, an Honors Bachelor of Arts in International Studies, and a minor in Business Administration; a Masters' of Science in Mechanical Engineering; and a Doctor of Philosophy in Mechanical Engineering with a minor in Industrial Engineering—all from Oregon State University.

(E-mail address: douglas.vanbossuyt@nps.edu)