INSTITUTE FOR DEFENSE ANALYSES

DevSecOps: State of the Art and Relevance to the Department of Defense

Jonathan R. Agre, *Project Leader*

June 2020

IDA Non-Standard
NS D-13199

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

The Institute for Defense Analyses is a nonprofit corporation that operates three Federally Funded Research and Development Centers. Its mission is to answer the most challenging U.S. security and science policy questions with objective analysis, leveraging extraordinary scientific, technical, and analytic expertise.

About This Publication

This work was conducted by the IDA Systems and Analyses Center under contract HQ0034-14-D-0001, Project ITSDPB, "ITSD Publications," for the IDA. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments

Karen D. Gordon

For More Information

Jonathan R. Agre, Project Leader
jagre@ida.org, 703-933-6522

Margaret E. Myers, Director, Information Technology and Systems Division
mmyers@ida.org, 703-578-2782

Rigorous Analysis | Trusted Expertise | Service to the Nation

INSTITUTE FOR DEFENSE ANALYSES

IDA Non-Standard NS D-13199

DevSecOps: State of the Art and Relevance to the Department of Defense

Jonathan R. Agre, *Project Leader*

DevSecOps: State of the Art and Relevance to the Department of Defense

Jonathan R. Agre

05/29/2020

INSTITUTE FOR DEFENSE ANALYSES

Abstract: Development-Security-Operations (DevSecOps) is an agile software development method in which developers, operators, and security experts are part of an integrated team responsible for the entire life cycle of a software product. DevSecOps can help solve many issues with traditional software methods, which often result in obsolete requirements and late discovery of security flaws after the software is delivered. The Department of Defense is interested in using this method to more rapidly develop and field secure software.

1. Introduction

Software development continues to move away from the traditional waterfall model toward more agile methods that reduce the development time and seek to ensure the end product better meets the customer's requirements. This has been reflected in new development-operations (DevOps) methods that encourage close cooperation between the software developers and the system operators by forming an integrated team composed of both developers and operators. DevOps is based on three important principles:

- Developers and operators are part of a single team that owns the repeating software life cycle from development to operation to development and so on.
- Automation of the development process— including testing, monitoring, and bug reporting—through tools that reduce the manual effort required for these activities.
- A continuous integration/continuous delivery (CI/CD) cycle that enables continuous improvements and rapid delivery of software versions to operations, sometimes several times per day.

The DevOps software development effort is typically divided into smaller development stages or versions, and automated tools allow frequent, rapid iteration of development and testing as the software product versions are deployed. Monitoring and operator feedback are injected directly back to the developers without delay. Together, this results in a continuous cycle.

DevSecOps arose from the realization that security considerations were slowing down the overall process. This resulted in the concept of introducing security personnel into the DevOps team during both development and operations so that security concerns and security requirements are addressed during the entire DevOps life cycle.

This paper covers basics about DevSecOps; its benefits to software development, especially for DoD; obstacles to implementation; the state of the art; and the market. The paper concludes by summarizing open issues and potential application areas.

2. Overview of DevSecOps

DevSecOps has evolved from agile software methods and DevOps concepts incorporating ideas from both software methodologies. It attempts to address the well-known problems of security flaws in software and software that is outdated by the time it is delivered due to cumbersome traditional software development processes.

2.1. DevOps Methods

As shown in Figure 1, DevOps methods are a continuous cycle of the following basic stages: Plan, Create (Develop), Verify, Preproduction, Release, Configure, and Monitor. Software systems are divided into smaller deliverable units (minimal viable product versions) that are moved all the way through to production in rapid bursts; many such units can be in the pipeline concurrently. In a DevOps environment, the requirements are not comprehensive and are often cast as features—usually restricted to about one page of textual description. The DevOps stages are executed continually, so that the software development is more of an evolutionary process rather than a discrete set of major releases produced over longer periods of time and delivered in larger blocks. A successful DevOps environment relies on many software development tools to be integrated into a toolchain that automates each step as much as possible. This results in continuous improvement through a cycle of continuous integration of software changes followed by a continuous delivery of modified software to operations; all stages are facilitated by automated processes and monitoring to ensure rapid responses to discovered bugs, and security vulnerabilities are fed back to the developers. Many large companies are reporting phenomenal success after adopting DevOps methods. For example, Amazon reported deploying code every 11.7 seconds on average, and Netflix does so thousands of times per day.[1]

A DevOps environment toolchain is a collection of tools that are integrated into development and operations to automate the functions needed at each stage to reduce time and effort. As shown in Figure 1, there are a variety of commercial and open source tools that can be put together to create an effective DevOps toolchain. A code repository (e.g., GitHub), a build tool (e.g., Jenkins), and release tools (e.g., Atlassian) are examples of important components. Some tool sets are used in several stages such as Atlassian and Red Hat Ansible. An important function not called out in the figure is bug/issue tracking such as that provided by Atlassian JIRA—this should also be integrated into the toolchain.

---

[1] Null, Christopher, "10 Companies Killing It at DevOps," *TechBeacon.* https://techbeacon.com/devops/10-companies-killing-it-devops

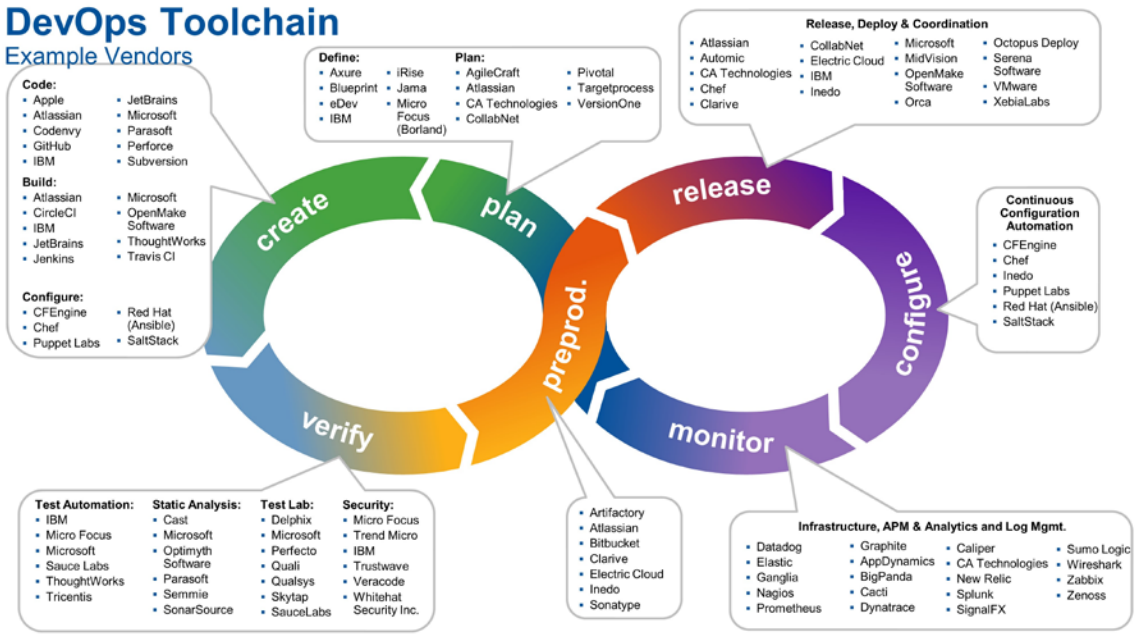

Figure 1: DevOps Toolchain[2]

2.2. DevSecOps Methods

In recognition of the difficulties involved with producing secure software, DevSecOps fully incorporates security considerations into each step of the entire DevOps process. DevSecOps uses a single team of developers, security experts, and operators for each software product. Traditional application security approaches rely on extensive testing and inspection, which typically occur late in the development process, often take weeks, and rely on separate groups of security professionals. In DevSecOps, security is considered from the beginning of the development process when security decisions are likely to have the largest impact on time and cost. It allows security tradeoffs to be inputs to design and development decisions throughout the continuous cycle.

Two central ideas in DevSecOps are (1) to enable and enforce a consistent implementation of security standards and (2) to automate most of the security compliance and testing functions to reduce the time required to find and fix security issues. Security is reassessed at each stage and with each iteration, using continuous risk management as input to decisions. In this approach, it is not necessary to entirely reduce the security risk during the development process, but a risk analysis can allow for runtime security controls in the overall risk assessment, ensuring that flaws

[2] Smith, David, "How to Build a Could Toolchain in DevOps," *HostReview*, Oct. 15, 2019. https://www.hostreview.com/blog/191015-how-to-build-a-cloud-toolchain-in-devops

discovered during operations are quickly fixed at the next iteration and the corrected code deployed.[3,4]

Figure 2 identifies some of the security tool types and functions used in a DevSecOps toolchain. According to Gartner, "many of the tools that have traditionally supported the secure software development life cycle (secure SDLC) will be applicable in DevSecOps, although the speed and frequency with which they run may be very different. In some cases, new tools will need to be purchased to address gaps or when the vendor doesn't modify its capabilities to be suitable for automation (for example, API enablement)."[5] We believe the monitoring and analytics activities are crucial for obtaining feedback from both the development and operations activities. Many of these security tools—which include some commercial and some open source tools—have long been used in traditional application security testing environments and are readily available; however, they may need to be adapted to operate at the quicker pace expected in the DevSecOps cycles. Throughout the cycle, application of the testing tools should be automatically invoked by the toolchain and the results integrated with the bug/issue reporting system with quick feedback and guidance to the developers.

---

[3] Defense Innovation Board, "Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage", May 3, 2019, *Security Accreditation*, pg. S101-S102. https://innovation.defense.gov/software/

[4] Serbu, Jared, "Air Force to Release New 'Fast-Track' Cyber Approval Process," Dec. 11, 2018. https://federalnewsnetwork.com/dod-reporters-notebook-jared-serbu/2018/12/air-force-to-release-new-fast-track-cyber-approval-process/

[5] Gartner "Integrating Security into the DevSecOps Toolchain", Mark Horvath, Neil MacDonald, Nov. 15, 2019.

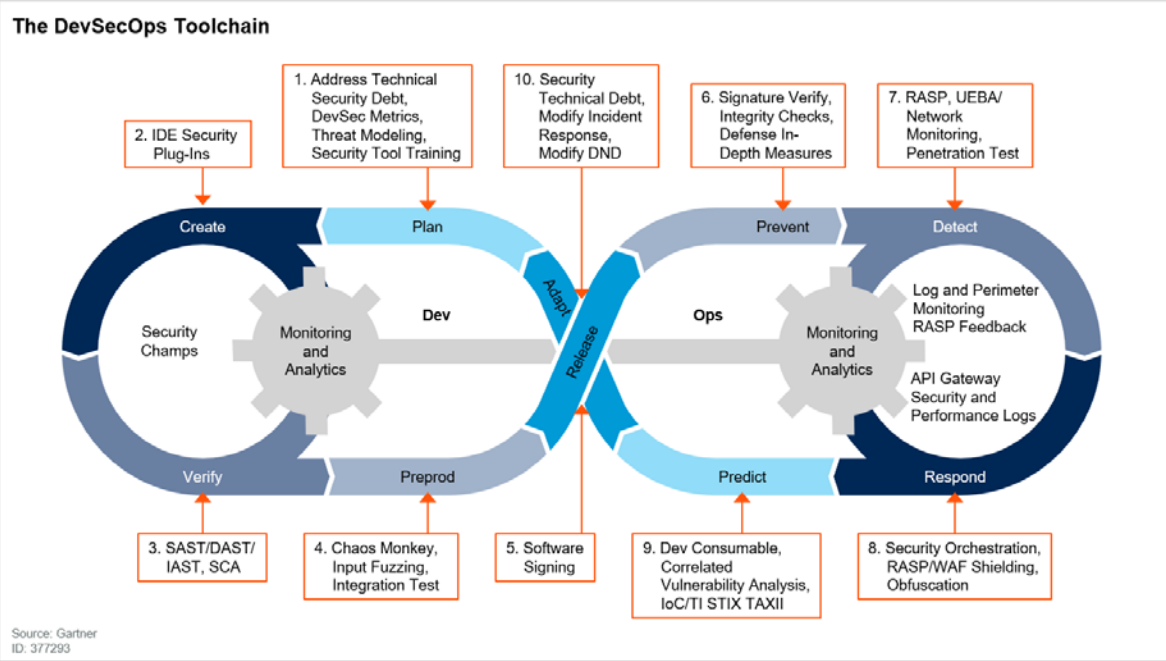

Figure 2: DevSecOps Toolchain[6]

DevSecOps requires clear security goals and threat models, as well as knowledgeable developers and security experts, to integrate adequate security and compliance testing. A successful DevSecOps process should result in a reduced security approval process, as the application would be built to meet security requirements and should not require time-consuming redesigns or fixes after release.

To streamline the security approval process, a development team must:

- Ensure that security controls are testable, largely through automated testing integrated into the toolchain. The application and the environment should be well instrumented to allow monitoring and analysis of the activities. This enables a developer to quickly validate security properties at each build, making it continuous and traceable.
- Adopt practices that utilize reciprocity and inheritance, and require developers to build applications starting with known good solution modules or libraries with inherited security properties that have already been validated and approved for use whenever possible. This can limit choices for developers but greatly simplifies testing by allowing security approvers to focus their attention on the newly developed code.
- Validate and approve the DevSecOps process itself—primarily, the components of the toolchain and handling of the artifacts. By developing and operating applications within an approved process, an approver can focus on the newly developed code rather than the

[6] Gartner "Integrating Security into the DevSecOps Toolchain", Mark Horvath, Neil MacDonald, Nov. 15, 2019.

infrastructure. In particular, container technologies[7] such as Docker[8] and Kubernetes[9] are used to create approved software components and play a prominent role in creating approved toolchains and reduced approval cycles.

- Use best practices, such as signing and verifying all developed software, the software tools, configuration files, and scripts.

2.3. Benefits of, and Obstacles to, DevSecOps

The benefits of adopting DevSecOps include those derived from the underlying DevOps process, as well as those from the embedded security expertise and built-in automated security verification:

- Reduced overall time to production of operational products. Although the process may deliver capabilities in smaller increments, the time to achieve operational status is reduced. Given the rate of change of the operational requirements of most software products that are delivered to the users, there is great advantage in quickly achieving an operational capability.
- Increased deployment frequency resulting from the smaller software modules developed at each iteration. Practitioners can deliver new versions that better respond to user needs and changing security threat knowledge.
- Cost reduction from detecting and fixing bugs and security vulnerabilities earlier in the development phase when it is cheaper to modify the code and from only reviewing the small amounts of code modifications in the incremental updates from the previous iteration.
- Increased security through automation of scanning and testing tools that provide consistent testing of any modifications during development. The rapid reporting of discovered security flaws through the embedded operators and automated monitoring results in reduced recovery times. Some experts claim that the reduced cycle time from release to update will result in more secure software simply due to the increased frequency of testing and validation and the quick turn-around from constant monitoring and reporting.[10]
- Management of the security auditing, monitoring, and notification through the DevSecOps workflows. This includes enforcing regular and automatic software patching that is integral to the process.
- Embedded automated risk characterization and monitoring at all stages of development, which keeps the products and environment in compliance.

---

[7] A *container* is a form of application virtualization that is implemented using the operating system running on the host computing platform. This is in contrast to *virtual machines*, which are implemented using a hypervisor on the host.

[8] Docker is an open-source standard for creating containers.

[9] Kubernetes is an open source tool for orchestration of a collection of containers. Kubernetes manages the details of deploying and monitoring containers.

[10] *Defense Innovation Board, Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage*, May 3, 2019, pg. viii. https://innovation.defense.gov/software/

- Use of a certified process, reusable tools and libraries, and standard infrastructure to reduce approval times to obtain permission to deploy and operate an application. This relies on the "secure by design" principle that is enforced by the process.
- Greater transparency and communication among the team members resulting in shared responsibility for security from the initial requirements throughout the life cycle of the product. Automatic metric and bug reporting that is visible to the entire team facilitates the rapid solving of issues.

While offering many benefits, DevSecOps may result in substantial organizational challenges to software development organizations due to the change in culture required to implement the DevSecOps method:

- Very large software initiatives may be difficult to break into smaller pieces and then manage and integrate.
- Application projects incorporating legacy software may not be amenable to this process.
- The typical separation between the developer teams and the operators needs to be eliminated to combine the teams and encourage communication.
- It is difficult to put together a team with the right mix of skills and experience. The staff must be trained with DevSecOps concepts and technologies to gain buy-in from the participants.
- The environment must produce actionable status information, such as security alerts or quality assurance reports and metrics, and make them available to the team members to allow collaboration.
- The policies governing security approvals must be modified to permit rapid accreditation of software versions.
- Cost estimation is difficult for DevSecOps due to the responsive and continuous nature of the process. In general, the application life cycle should be funded continuously for as long as the application is fielded.

Despite these difficulties, DevSecOps seems to be a promising solution for meeting the needs of rapid secure software delivery to customers in many cases.

2.4. Metrics for DevSecOps

The DevSecOps process presents additional challenges due to the differences from traditional methods. The process must be managed and monitored in a way to capture how well it functions and to reflect the goal of rapid delivery of useful software. Traditional software metrics such as "source lines of code" are known to be poor indicators of software progress and quality, and they are not appropriate in this environment. Operational metrics such as help desk tickets and availability are important but also need to be understood in the context of DevSecOps. DevSecOps should use metrics such as speed of delivery, cycle time, security, quality of code, and useful capability delivered. As an example, the Defense Innovation Board's Software Acquisition and Practices (SWAP) report has suggested a set of 14 metrics that reflect the

objectives of a DevSecOps environment and illustrate the differences from traditional metrics (see Appendix A).[11]

3. DoD Adoption of DevSecOps

The DoD's recent interest in DevSecOps was triggered by recommendations in the May 2019 SWAP study in which DevSecOps is put forward as the best way to deliver software at the speed required to maintain and protect critical DoD systems.[12] The report included specific recommendations for addressing many of issues facing software development by the DoD; the recommendations were grouped into four categories: (1) policy and statutes, (2) creating shareable common digital infrastructure, (3) improving the workforce, and (4) adopting DevSecOps.

In July 2019, the Under Secretary of Defense for Acquisition and Sustainment (USD(A&S)) and the DoD Chief Information Officer (CIO), together with the DoD Services, established the DoD Enterprise DevSecOps Initiative as a joint program, and they have been conducting an active working group exploring the recommendations from the Defense Innovation Board's SWAP report.[13] In August 2019, the DoD CIO published the DevSecOps reference design, describing an architecture for a DevSecOps environment and providing guidance for stakeholders such as developers, operators, and approvers.[14] A joint USD(A&S) and DoD CIO signed memo was released in October 2019, making DevSecOps the preferred software development methodology for DoD.[15] In January 2020, a new Software Acquisition Pathway was introduced by the USD(A&S) that streamlined software acquisition, formally defined the preferred path for software development, and firmly endorsed the principles of DevSecOps.[16]

As a further sign of endorsement, the Director, Operational Test and Evaluation, recently stated:

> Repeatable automated testing will reduce man-hours required for testing system changes and enable delivery of software at the speed of relevance. It will enable evaluating the effect system changes or failures have on the safety and capabilities of the warfighter. Repeatable automated testing will improve system sustainability and cost through early detection and resolution of deficiencies. To facilitate these improved software development considerations, the DOD should implement an iterative, incremental approach to acquisition and T&E, such as Development

---

[11] *Defense Innovation Board, Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage*, May 3, 2019. https://innovation.defense.gov/software/

[12] Ibid

[13] "DoD Enterprise DevSecOps Community of Practice (CoP)," Briefing, Sept. 9, 2019.

[14] DoD Chief Information Officer, *DoD Enterprise DevSecOps Reference Design, Version 1.0*, Aug. 12, 2019.

[15] Deasy, Dana and Ellen Lord, "Software Development, Security, and Operations for Software Agility," DoD Memo, Oct. 24, 2019.

[16] Lord, Ellen M., "Software Acquisition Pathway Interim Policy and Procedures," Memorandum, January 3, 2020.

> Security Operations (DevSecOps). During DevSecOps, stakeholders (i.e., system developers, acquirers, developmental and operational testers, cybersecurity experts, and warfighters) collaborate across the entire system lifecycle, from development and test to operations and sustainment.[17]

3.1. DevSecOps for the DoD

The DevSecOps concept, as described in the DoD reference design report, is shown in Figure 3. The reference design relies heavily on the notion of security-hardened containers to implement the toolchain. It envisions the creation of a software factory service, utilized by DoD entities, that will produce the security-hardened containers for the software tools and developed applications that satisfy DoD security requirements. More specifically, the report describes a DevSecOps software factory for producing Open Container Initiative[18] containers that can be run in a Kubernetes-orchestrated environment that meet the security requirements. A DoD Enterprise DevSecOps initiative has enlisted a software development team from the United States Air Force (USAF) that has implemented an example of a software factory to provide hardened DevSecOps containerized tools and has produced over 170 containerized components for use in building additional DevSecOps software factories.[19] These can be used to replicate and tailor a software factory at other installations with the goal of quickly approving the new DevSecOps process.

The USAF has been successfully piloting DevSecOps in its Kessel Run software factory based on teaming with Pivotal, a small software development company.[20] In addition, the Air Force is implementing several DevSecOps pilot initiatives to speed up adoption: Cloud One, a standardized cloud infrastructure (on AWS and Azure) with baked-in security and an authority to operate (ATO); and Platform One by LevelUP, a program to rapidly develop cyber software using a Kubernetes-compliant containerized platform with a CI/CD pipeline.[21] Platform One allows development teams to choose among a variety of hardened containerized tools and environments to customize their CI/CD process. Platform One also assists other DoD entities by providing a pay-per-use "DevSecOps as a Service" environment.[22]

---

[17] Behler, Robert, Director, *Operational Test and Evaluation FY2018 Annual Report*, Dec. 2018.

[18] The Open Container Initiative is a standard for specifying Docker images and their runtime environment. https://www.opencontainers.org/

[19] Nicolas Chaillan, "DoD Enterprise DevSecOps Initiative (Software Factory)," v1.5, Briefing, Nov. 18, 2019. https://pt.slideshare.net/scoopnewsgroup/devsecops-the-dod-software-factory

[20] Mark Pomerleau, "How the Air Force's New Software Team Is Proving Its Worth," *C4ISRNet*, Jan. 19, 2019. https://www.c4isrnet.com/it-networks/2019/01/14/how-the-air-forces-new-software-team-is-proving-its-worth/

[21] Nicolas Chaillan, "DoD Enterprise DevSecOps Initiative (Software Factory)," v1.5, Nov. 18, 2019. https://pt.slideshare.net/scoopnewsgroup/devsecops-the-dod-software-factory

[22] Platform One: DoD Enterprise DevSecOps Services, https://software.af.mil/dsop/services/.

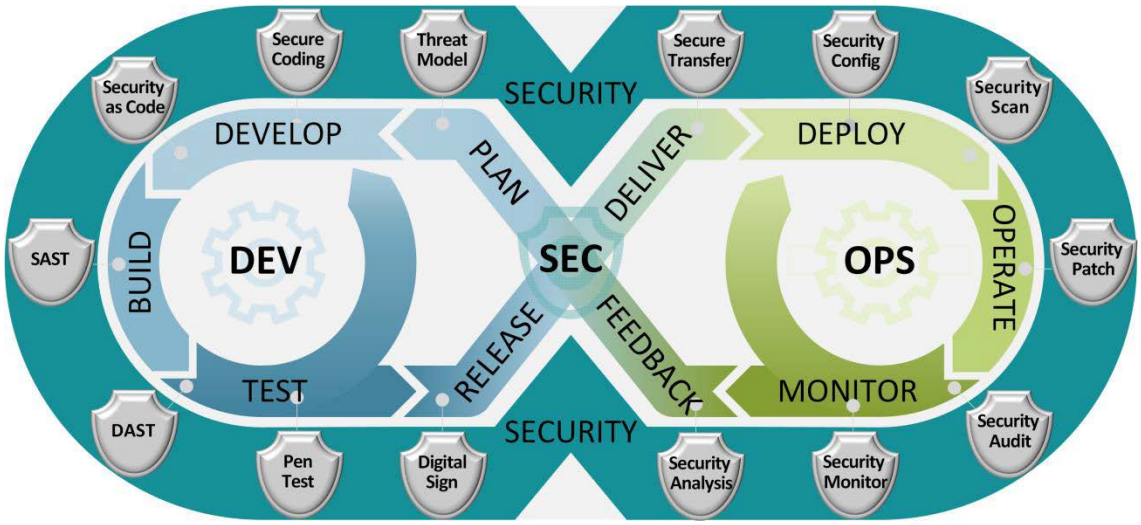

Figure 3: The DoD DevSecOps Reference Design

3.2. Continuous Authority to Operate (c-ATO)

The DoD is also increasingly interested in reducing the time required to obtain an ATO— a security accreditation and permission to deploy the application on DoD systems. Obtaining an ATO is a time-consuming process and can take several months to years for large software projects. A key requirement is that a development environment or an application is required to be compliant with the controls defined by the NIST 800-37 Risk Management Framework (RMF).[23] As pointed out in the SWAP report, this is often accomplished by a cursory, manual process and does not necessarily result in increased security. Further, if the time to receive approval is greater than the time to produce new versions of an application, this process will never be satisfactory. Automation of the implementation and testing of controls such as RMF is an important part of the DevSecOps process.

To further address this problem, the SWAP report recommended that the DoD begin to accredit the DevSecOps process used to produce the software rather than the individual applications and their development environments. They also introduced the concept of a continuous ATO (c-ATO) that relies heavily on an accredited DevSecOps environment. As long as the process stays accredited and there is continuous monitoring of the operations, then an authorizing official can approve a c-ATO. In a c-ATO, the approver needs only additionally review any modified portions of the environment. If it is based on a standard component, then the approvals should be straightforward. The SWAP report recommends standardization of tools, software libraries, and best practices to maximize the ability to inherit the approvals of previously vetted and approved items in order to reduce the time needed to approve the process.

---

[23] NIST, Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy, *NIST Special Publication SP 800-37 Rev. 2, Dec. 2018.* https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final

Platform One by LevelUP is intended to help support c-ATO and has reported achieving more than 10 updates per day.[24] The Kessel Run software factory has reportedly been granted a c-ATO.[25] There have been several other pilot efforts to implement continuous or reduced time ATOs:

- The General Services Administration (GSA) introduced their 18F process, which reduced their time to complete an ATO process from more than six months to under a month.[26]
- The National Geospatial-Intelligence Agency (NGA) ATO-in-a-Day Initiative leverages automation in conjunction with standardized infrastructure, components, and processes to drastically reduce the approval time, with the eventual goal of approvals in one day.[27]
- Air Force Fast-Track ATO and Ongoing Authorization is a path within the Air Force Risk Management Framework that allows an authorization decision based on a combination of a cybersecurity baseline, an assessment, and an information systems continuous monitoring strategy.[28] The Fast-Track ATO relies on conducting assessments using penetration testing and enables the ATO process to be completed in five weeks as opposed to the typical year or more.[29,30]

Commercial cloud providers such as Amazon AWS and Microsoft Azure are beginning to assist in setting up environments that will facilitate c-ATOs.[31,32] The cloud providers have developed services and templates that include continuous monitoring services and automated evaluation of RMF controls. This helps generate the documentation needed for FedRAMP and DoD ATO's.

---

[24] Aaron Boyd, The Air Force's Platform One Team Thought It Was Agile. Then COVID-19 Hit.,NextGov, May 27, 2020. https://www.nextgov.com/emerging-tech/2020/05/air-forces-platform-one-team-thought-it-was-agile-then-covid-19-hit/165676/.

[25] Marc Pomerleau, How the Air Force's new software team is proving its worth, C4ISRNET, Jan. 14, 2019. https://www.c4isrnet.com/it-networks/2019/01/14/how-the-air-forces-new-software-team-is-proving-its-worth/

[26] Aidan Feldman, "Taking the ATO Process from 6 Months to 30 Days," July 19, 2018, https://18f.gsa.gov/2018/07/19/taking-the-ato-process-from-6-months-to-30-days/.

[27] Jason Miller, "GSA, NGA Shrink Time to Cyber-Approve Systems from Year to Month," Aug. 6, 2018, https://federalnewsnetwork.com/reporters-notebook-jason-miller/2018/08/gsa-nga-shrink-time-to-cyber-approve-systems-from-year-to-month/

[28] SAF/CN, Memorandum for Authorizing Officials, "Fast-Track Authorization to Operate (ATO)," 18 March 2019. This memorandum was accompanied by a cover memorandum from the Under Secretary of the Air Force, dated 22 March 2019, to all Major Command (MAJCOM) Commanders and Authorizing Officials.

[29] Billy Mitchell, "Cybersecurity ATOs Faster: Air Force Set up New Fast Track," *FedScoop*, April 22, 2019, https://www.fedscoop.com/fast-track-ato-air-force-wanda-jones-heath/.

[30] Boyd, Aaron, "Air Force's New Fast-Track Process Can Grant Cybersecurity Authorizations in One Week," Mar. 27 2019, https://www.nextgov.com/cybersecurity/2019/03/air-forces-new-fast-track-process-can-grant-cybersecurity-authorizations-one-week/155860/

[31] https://www.slideshare.net/AmazonWebServices/the-quest-for-continuous-ato-a-case-study-featuring-the-us-intelligence-community

[32] "Risk Management Framework Compliance: Introducing ATO as a Service," https://cfocussoftware.com/risk-management-framework/

4. DevSecOps State of the Practice

Given the momentum of DoD and others toward adopting a DevSecOps environment, a quick look at the state of the practice is warranted. Gartner estimates, "By 2021, DevSecOps practices will be embedded in 60% of rapid development teams as opposed to 20% in 2019. By 2023, more than 70% of enterprise DevSecOps initiatives will have incorporated automated security vulnerability and configuration scanning for open-source components and commercial packages, which is a significant increase from fewer than 30% in 2019."[33]

In the Magic Quadrant for Application Security Testing report, Gartner states "modern application design and the continued adoption of DevSecOps are expanding the scope of the AST market. Security and risk management leaders will need to meet tighter deadlines and test more complex applications by seamlessly integrating and automating AST in the software delivery life cycle."[34]

In "Hype Cycle for Application Security, 2019,"[35] Gartner has positioned technologies related to application security testing according to their maturity in the market. DevSecOps is positioned to be near the end of the Trough of Disillusionment and should achieve market maturity in 5 to 10 years. We believe that static and dynamic application security testing are typically part of an Application Security Testing Suite, which is positioned in the Slope of Enlightenment. Interactive Application Security testing is still in an earlier state, located on the Trough of Disillusionment. Some other key technologies for DevSecOps—Application Monitoring and Protection and API Security Testing and Discovery—are still considered to be in the Innovation Trigger phase and are 5 to 10 years out from mainstream adoption. Other technologies likely to impact DevSecOps that are positioned in the Peak of Inflated Expectations phase are Software Usage Analytics, expected to mature in 5 to 10 years, and Application Security Orchestration and Correlation, expected to mature in 2 to 5 years.

As indicated above, certain DevSecOps technologies are still in an early stage; however, most of the tools needed to implement the methodology are available as commercial off-the-shelf (COTS) or open source products. Some of the automated testing technologies and tools are still emerging from research and development efforts, so some manual intervention will still be required to implement the entire DevSecOps life cycle.

5. Open Issues

DevSecOps methods are still evolving, and there are many open issues. As mentioned earlier, not all applications easily fit into this method. Several particular areas of concern are described below.

---

[33] Gartner "12 Things to Get Right for Successful DevSecOps," Neil MacDonald, Dale Gardner, Dec. 19, 2019.

[34] Gartner "Magic Quadrant for Application Security Testing," Mark Horvath et al., 29 April 2020

[35] Gartner "Hype Cycle for Application Security, 2019," Mark Horvath, 30 July 2019

Testing Environment – In general, the DevSecOps environments have not adequately addressed testing in a global, distributed environment, although there has been success with applications that operate in specific, limited domains. If the application is widely distributed, or has need to access remote resources, it may be difficult to create the testing scenario in an accredited environment. For example, some cyber security tools require specialized cyber testing ranges to determine if they are functioning correctly and to discover potential security flaws. Applications such as distributed content management may also present difficulties in replicating an environment of sufficient fidelity. There are additional challenges with implementing distributed and highly secure systems in a DevSecOps environment.[36]

Integration of Simulation – These has been little discussion of the use of simulated environments for testing purposes. Simulation models should be easily automated and integrated into the toolchain. The challenge would be to determine and measure the risk involved with using simulated environment components.

Mobile Applications - Mobile applications include native apps, hybrid apps, progressive web apps, and traditional web apps. These apps are typically characterized as small applications with short development times and frequent updates and seem especially appropriate for DevSecOps. There are many specialized tools for mobile application development and testing. Can these tools be hardened and used in a DevSecOps environment? Is the high frequency of updates common with mobile apps sustainable in the environment, in particular with a c-ATO? Are containers the appropriate method for mobile tools? Can the mobile environment be replicated and adequately tested?

Cloud Implementations – There are still ongoing discussions about how the DoD could accredit a DevSecOps environment in a commercial public cloud. If one of the approved GovCloud environments is employed, then the environment would inherit those security features, but in a commercial public cloud, additional security methods would need to be implemented and responsibilities allocated. Use of DoD-vetted containers should help with approvals.

Container Security – Containers and container systems present a host of security challenges. NIST has published an application security guide that describes many of the pitfalls and difficulties in securing container environments.[37] However, the DoD is recommending extensive use of containers in their DevSecOps environments. Monitoring of the potential security of containers needs to be an ongoing process.

Instrumentation and Monitoring – Instrumentation of the DevSecOps environment and the applications operating in the field is necessary to properly evaluate how well the process is

---

[36] Jose Morales et al, "Guide to Implementing DevSecOps for a System of Systems in Highly Regulated Environments," Software Engineering Institute, Technical Report CMU/SEI-2020-TR-002, April 2020.

[37] Souppaya, Murugiah P., John Morello, Karen A. Scarfone, *Application Container Security Guide*, NIST Special Publication (NIST SP) - 800-190, Sept. 25, 2017. https://www.nist.gov/publications/application-container-security-guide

functioning. Converging and standardizing on a set of common metrics would allow cross-environment comparisons and help with automating those metrics. Appropriate sensors to gather the data and analytical tools to evaluate the newer metrics still need to be developed for DevSecOps processes and applications.

6. Conclusions

DevSecOps is emerging as the most promising method of delivering modern software at the required speed and quality. For the DoD, in particular, this speed is needed to assure that the software in their enterprise and weapon systems continues to be updated to meet the changing threats. A DevSecOps process can be built around a software factory that uses standardized, approved toolchain components of hardware and software and that would produce applications that qualify for a c-ATO. The ability to obtain a c-ATO relies on (1) a development process that inherits security controls by using standardized components and (2) automation of testing and monitoring for vulnerabilities and compliance. Existing pilot programs, such as Kessel Run, could be used to create and deploy these DevSecOps software factories across the DoD to verify they will scale across the variety of DoD software systems.

It is important for the DoD to stand up more DevSecOps environments to gain hands-on experience with the process and identify and correct the institutional barriers to its success. In addition, the DoD should continue to observe developments in the research and academic software engineering communities and adopt the emerging tools and best practices relating to DevSecOps. Given the rapid commercial adoption and successful DoD pilot activities, DevSecOps is expected to become the standard rather than the rare experiment within the next several years.

Appendix A: Software Metrics

The following table is adapted from Appendix E of the SWAP report.[38] The metrics and target values are shown for several different types of software, as are the typical values for software as currently developed by the DoD. The target values are notional. The metrics are further categorized as:

- Deployment Rate - # 1,2,3
- Response Rate - # 4,5
- Code Quality - # 6,7,8,9
- Functionality - # 10,11
- Program Management, Assessment, and Estimation - # 12,13,14

#	Metric	Target Value (by software type)				Typical DoD values for SW
		COTS Apps	Customized SW	COTS HW/OS	Real-time HW/SW	
1	Time from program launch to deployment of simplest useful functionality	<1 mo	<3 mo	<6 mo	<1yr	3 yrs
2	Time to field high priority fcn (spec->ops)	N/A	<1 mo	<3 mo	<3 mo	1-5 yrs
	Fix newly found security hole (find->ops)	<1 wk	<1 wk	<1 wk	<1 wk	1-18 mo
3	Time from code committed to code in use	<1 wk	<1 hr	<1 day	<1 mo	1-18 mo
4	Time required for full automated regression test	N/A	<1 day	<1 day	<1 wk	2 yrs
	Cybersecurity audit/penetration testing	<1 mo	<1 mo	<1 mo	<3 mo	2 yrs
5	Time required to restore service after outage	<1 hr	<6 hr	<1 day	N/A	?
6	Automated test coverage of specs/code	N/A	>90%	>90%	100%	?
7	Number of bugs caught in testing vs field use	N/A	>75%	>75%	>90%	?
8	Change failure rate (rollback deployed code)	<1%	<5%	<10%	<1%	?
9	% code avail to DoD for inspection/ rebuild	NA	100%	100%	100%	?

[38] Defense Innovation Board, "Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage," May 3, 2019. https://innovation.defense.gov/software/

10	Number/percentage of functions implemented	80%	90%	70%	95%	100%
11	Usage and user satisfaction	TBD	TBD	TBD	TBD	?
12	Complexity metrics	#/type specs # programmers Structure of code #/skill level of teams #/type of platforms #/type deployments				Partial/ Manual tracking
13	Development plan/environment metrics					
14	Nunn-McCurdy threshold (for any metric)	1.1x	1.25x	1.5x	1.5x per effort	1.25x total $

This work was conducted under contract HQ0034-14-D-0001, Project ITSDPD, for professional development. The publication of this IDA memorandum does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency. The material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7013 (a)(16) [Jun 2013].

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From – To)
00-06-20	Non-Standard	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
DevSecOps: State of the Art and Relevance to the Department of Defense	HQ0034-14-D-0001
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBERS

6. AUTHOR(S)	5d. PROJECT NUMBER
Jonathan R. Agre	ITSDPB
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES	8. PERFORMING ORGANIZATION REPORT NUMBER
Institute for Defense Analyses 4850 Mark Center Drive Alexandria, VA 22311-1882	NS D-13199

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR'S / MONITOR'S ACRONYM
Institute for Defense Analyses 4850 Mark Center Dr., Alexandria, VA 22311	IDA
	11. SPONSOR'S / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Project Leader: Jonathan R. Agre

14. ABSTRACT

Development-Security-Operations (DevSecOps) is an agile software development method in which developers, operators, and security experts are part of a single team responsible for the entire life cycle of a software product. DevSecOps can help solve many issues with traditional software methods that result in obsolete requirements and late discovery of security flaws after the software is delivered. The Department of Defense is interested in using this method to more rapidly develop and field secure software.

15. SUBJECT TERMS

DevSecOps, DevOps, application security, toolchain, software development, software test, authority to operate

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Institute for Defense Analyses
Unclassified	Unclassified	Unclassified	Unlimited	17	19b. TELEPHONE NUMBER (Include Area Code)