Journal of the KNST

2022; Vol.5, No.2; pp. 122-127

ISSN: 2635-4926

https://doi.org/10.31818/JKNST.2022.09.5.2.122

Check for updates

Received: 2022/08/27
Revised: 2022/09/08
Accepted: 2022/09/28
Published: 2022/09/30

*Corresponding Author:
Kyungwon Oh
TEL: +82-63-450-7724
E-mail: oh.kyungwon@gmail.com

Abstract

본 논문은 국방로봇 소프트웨어 개발에 DevSecOps 프레임을 적용하는 방법을 제안한다. DevOps는 소프트웨어 개발 방법론으로 다양한 민수영역에서 적용되고 있으며, 최근 보안이 강화되어 개발/평가/실증을 같이 수행하며 안정적으로 개발을 수행하는 DevSecOps 방법으로 확장되어 적용되고 있다. 본 논문에서는 미 국방성의 DecSecOps 전략을 분석하고 이어서 국내 방산 환경에 적용하는 방법을 제안하였다. 제안된 소프트웨어 개발 방법은 국방로봇의 성능을 안정적·지속적으로 향상시킬 수 있다. 또한 유무인 복합체계(MUM-T)와 같이 높은 신뢰성을 담보하거나 역할을 분담하는 시스템 개발에 적용할 수 있는 방법을 제안하였다.

This paper proposes a method to apply the DevSecOps frame for defense robot SW development to overcome these existing development limitations. DevSecOps is a software development methodology that has been widely applied in the civil sector recently, and security at each stage is added to DevOps that performs development/evaluation/operation together to ensure a more stable development environment. In this paper, first, the application method and method of the existing DOD DevSecOps are analyzed, and secondly, a method to apply DevSecOps to the domestic environment is proposed. The proposed methodology can continuously improve the SW stability as well as the utilization of the existing defense robot. In addition, it can be applied to the robot development method for MUM-T (Man unmanned Team), which is emerging recently, and can be optimized for reliability formation and role sharing between users and unmanned robots.

Keywords

국방로봇(Military Robot), 유무인복합체계(MUM-T), DevOps, DevSecOps, 정보자율시스템(Intelligence Autonomous System)

국방로봇 SW 개발을 위한 DevSecOps 적용 방안

DevSecOps for Military Robot SW Development

김문환[1], 오경원[2*]

[1]LIG넥스원 해양연구소 수석연구원
[2]호원대학교 항공정비공학과 교수

Moon Hwan Kim[1], Kyungwon Oh[2*]

[1]Chief Research Engineer, Maritime Research Center, LIG Nex1 Co., Ltd.
[2]Professor, Dept. of Aircraft MRO Engineering, Howon University

1. 서론

최근 인공지능 및 제조 기술 발달에 따라서 다양한 무인체계 개발이 국방분야에서도 진행되고 있다. 일반적인 대부분의 무인체계가 로봇의 범주에 포함되기 때문에 국방로봇의 개발이 증가한다고 볼 수 있다. 국방로봇 개발은 하드웨어와 소프트웨어 개발로 분리하여 살펴볼 수 있고, 국내 하드웨어 개발 성숙도는 많이 증가되는 것에 반해 소프트웨어 개발은 아직 많은 연구가 필요한 실정이다.

로봇 소프트웨어 개발은 로봇 개발을 느리게 하는 가장 큰 원인 중 하나로, 어렵고 시간을 많이 필요로 하는 작업이다[1]. 이를 극복하기 위해 로봇 소프트웨어 개발을 가속하는 방법으로 개발환경, 시뮬레이션, 미들웨어 등을 사용하는 연구들이 이루어지고 있다[2]. 최근 많이 연구되고 있는 DevOps를 적용한 개발 방법은 높은 수준의 소프트웨어를 빠르게 배포하고 개발하여 소프트웨어 lifecycle 비용을 줄이고 있다[3].

국방로봇 소프트웨어 개발은 기존의 일반 상용 로봇 소프트웨어 개발보다 더 어려운 문제점을 가진다. 국방로봇 소프트웨어 개발은 무기체계 소프트웨어 개발 지침[4], 국방 아키텍처 프레임워크(MND-AF)[5], 국방 CBD 방법론[6]을 따라 무기체계 소프트웨어 획득 프로세스를 따라서 개발되어야 한다. 세부 프로세스 간 다양한 산출물을 통해 소프트웨어의 요구사항 추적과 규격화가 이루어지며 시험평가 간에 신뢰성 시험을 통해 소프트웨어 동작뿐만 아니라 코드 자체의 결함도 평가를 수행한다. 이러한 복잡한 관리절차는 요구사항을 만족시키고 결함을 줄여 소프트웨어 품질을 높이는 효과를 가진다. 따라서 이러한 접근법은 요구사항을 만족시키는 무기체계 개발에 적합한 방법이라고 할 수 있다.

무인체계로서 국방로봇의 가장 큰 특징 중 하나는 요구사항의 불명확성이다. 특히 MUM-T(manned-unmanned teaming) 개념에서 운용되는 국방로봇의 경우 운용자와 다양한 작전을 수행하면서 지속적으로 요구사항의 변경이 발생한다. 소요군 입장에서도 국방로봇의 명확한 요구사항의 확정하기 힘든 부분이 존재하고, 개발 이후 사용자 불만이 지속적으로 발생할 수 있다. 이러한 문제를 해결하기 위해 최근 미 해군에서 개발되고 있는 무인체계 ACTUV(ASW Continuous Trail Unmanned Vessel)의 경우는 시험 시제를 개발하고 시험 시제 운용을 통해서 요구사항을 구체화하여 2차 시제에 적용하는 형태로 개발하며 미 해군 Naval Sea System Command의 PMS 406에서는 DevSecOps 방식을 전체적인 무인체계 개발 방법으로 규정하고 있다[7].

본 논문에서는 새로운 무인체계 형태의 국방 로봇 소프트웨어 개발을 위해 새롭게 적용되고 있는 DevSecOps 개발 적용 방식을 제안한다. 미 국방부에서 적용하고 있는 DevSecOps 방법론을 분석하고 기존의 무기체계 소프트웨어 개발 방법과 비교하였다. 또한 기존 무기체계 개발 방법의 한계를 분석하였다. 2장에서는 DevOps 기반 소프트웨어 개발 방법을 논의하였으며, 3장에서 국방로봇 소프트웨어에 DevSecOps를 적용하는 방법에 대한 방법을 논의하였다. 마지막으로 4장에서 결론으로 논문을 마무리하였다.

2. DevOps 기반 소프트웨어 개발 방법

2.1 DevOps 개발 방법

DevOps는 development(개발)과 operations(운용)이 긴밀히 협조 연계하여 비즈니스 측면의 가치를 높이는 근무 방식과 문화를 말한다. DevOps는 애자일(agile) 개발에 의한 계속적인 개발로의 변화 및 계속적인 개발로 인해 나타나는 운용 과제로부터 기인하였다.

초기 소프트웨어 개발은 폭포수 방식을 따라서 개발이 되었다. 폭포수 개발 방법은 1980년대부터 서비스 개발에서 적용된 방법으로 계획, 요건 정의, 설계, 개발, 테스트, 배포 단계로 구성되어 있다. 폭포수 개발로 만들어진 서비스는 재개발의 개념이 없으며 한번 개발되면 장기간 서비스가 되는 것이 특징이다. 하지만 Web 기반 서비스가 발전함에 따라서 짧은 기간에 요구사항이 추가되거나 변경이 늘어가면서 서비스를 정책적으로 피드백 받아 개발을 여러 번 반복하는 프로토타입 모델 개발 기법이 등장하였고, 소규모 개발을 전제로 필요한 최소한의 요건을 적용한 성과물을 만들어 배포하고 고객의 피드백을 받아서 지속적인 개선을 반복하는 '애자일(Agile) 개발' 기법이 탄생하였다. 애자일 개발에서는 폭포수 개발과 달리 기능의 추가가 빈번하게 이루어지기 때문에 지속적 통합(CI, continuous integration)이라는 개념과 지속적 배포(CD, continuous delivery)의 개념이 도입되었다.

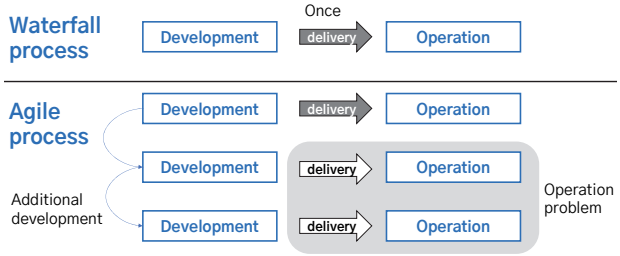

Fig. 1. Comparison between waterfall process and agile process

2.2 DevOps 개발 방법 관련 연구

DevOps 개발 방법 관련 연구는 대부분 국외에서 진행되고 있으며, 개발 문화에 대한 연구 방법이기 때문에 주로 설문 또는 통계적인 분석을 통해서 연구가 되고 있다.

Bauml 및 Hirzinger는 애자일 방식을 소프트웨어를 복잡한 로봇 시스템에 배포하여 개발하는 애자일 로봇 개발이라는 개념을 연구하였다[8]. 이 연구를 통해서 기존의 애자일 방식의 소프트웨어 개발 방법론이 로봇 소프트웨어 개발에 적용가능함을 확인하였다.

Brugali 와 Scandurra는 component 기반 소프트웨어 개발 방법(CBD)의 원칙과 구현 방법이 로봇 소프트웨어 개발에 적용 가능한지 연구하였다[9]. 이 연구를 통해서 각 로봇의 기능을 component 단위로 개발된 소프트웨어로 개발하여 적용 가능함이 확인되었다.

Bucena 와 Kirikova는 DevOps의 세부적인 적용 사례에 대하여 연구하였다[10]. 이 연구에서는 새롭게 DevOps 문화가 정착되면서 발생하는 도전적인 사항과 적용에 필요한 다양한 모델 및 도구들을 분석하였다.

Lwakatare 등은 각기 다른 5가지 개발 환경에서 DevOps 적용 사례를 분석하였다[11]. 다양한 크기의 회사, 다른 종류의 프로젝트에 DevOps를 적용한 사례가 분석되었다. 이 분석 결과 DevOps는 단순히 순수한 기술적인 개발 기법이 아니고 조직 문화와 참여 구성원의 의식

변화도 같이 필요하다는 결론을 도출하고 있다.

Wijava 등은 임베디드 시스템 개발에 DevOps를 적용한 사례를 분석하였다[12]. 특히 임베디드 시스템 개발 절차에 CI/CD 및 continous monitoring(CM)을 적용시켰다. 로봇 소프트웨어 개발이 대부분 임베디드 시스템에 적용되기 때문에 실제적인 로봇 소프트웨어 개발에 DevOps 개발 방법이 적용됨을 확인할 수 있다.

2.3 DevOps 개발 방법

DevOps 개발 방법은 여러 가지로 정의될 수 있지만 Fig. 2와 같이 크게 8가지 단계가 순환 반복되는 구조를 가진다. 각 단계 및 주요 활동은 다음과 같다.

(1) 계획: 문제를 분석하고 코딩을 계획한다.
(2) 코딩: 실제적인 코딩작업을 수행한다.
(3) 빌드: 코딩을 자동 빌드한다.
(4) 테스트: 코딩의 동작/결함을 테스트한다.
(5) 배포: 빌드가 완료된 코드를 배포한다.
(6) 배치: 배포된 코드를 각 서비스에 배치한다.
(7) 운용: 배치된 서비스를 운용한다.
(8) 모니터링: 운용하면서 코드의 성능을 모니터링한다.

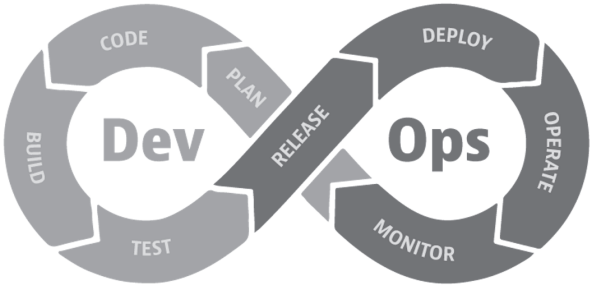

Fig. 2. DevOps process

2.4 DevSecOps 개발 방법

DevSecOps는 DevOps에 보안(security)을 추가한 방식이다. DevOps를 통해서 개발과 운용의 주기가 짧아지기 때문에 보안에 대한 이슈도 짧은 주기를 가지고 지속적으로 대응해야 한다. 따라서 기존의 DevOps의 운용 절차에 보안 절차 추가가 필요하다. DevSecOps를 통해서 안정적인 보안을 유지하며 빠른 개발/배포 주기를 가지는 시스템 구축이 가능하다.

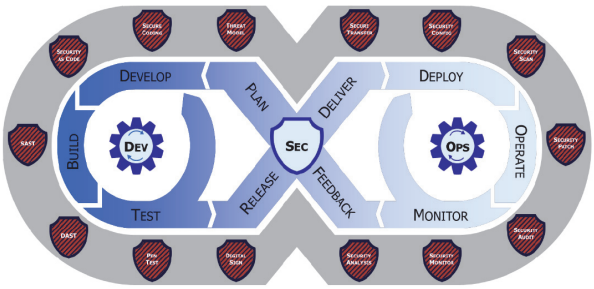

Fig. 3. DevSecOps process[17]

3. 국방로봇을 위한 DevSecOps 적용 방안

3.1 미 국방부의 DevSecOps 적용 사례

미국의 무인화 관련 연구는 1980년대부터 시작하였으며 2014년 이후 본격적으로 무인체계를 전력화하려는 노력을 수행하고 있다[14].

무인체계 전력화에서 핵심으로 떠오르는 것은 무인체계 소프트웨어 개발이다. 미 국방과학위원회(DSB, Defense Science Board)의 2011년 이후 발행된 무인화 관련 보고서에서는 무인화 시스템 개발의 핵심이 소프트웨어 개발임을 강조하고 있다[14]. Fig. 4는 지난 30년간의 소프트웨어 개발을 애플리케이션 수명주기, 개발 프로세서, 서버 인프라와 데이터베이스 관리 방법 그리고 보안 관련 모범사례의 변화를 나타낸다.

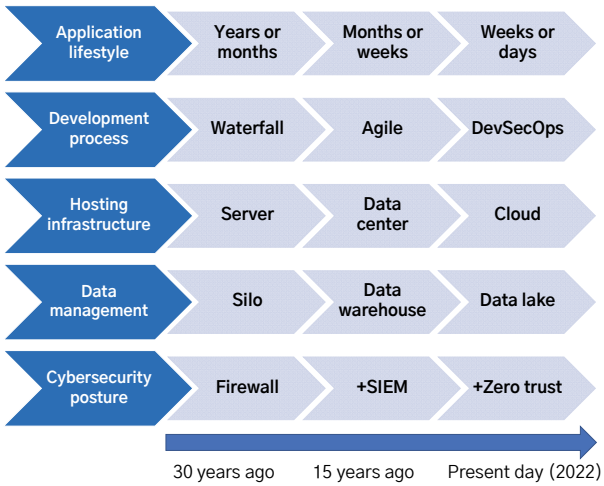

Fig. 4. Maturation of software development best practice [16]

소프트웨어 개발 수명주기는 연간에서 일간으로 단축되고 있으며, 개발 방식은 폭포수 방법에서 애자일 방법

으로, 이후 DevSecOps 방식으로 변화되고 있음을 알 수 있다. 데이터 관리를 위한 인프라의 경우도 서버에서 데이터 센터를 거쳐 클라우드로 발전하였으며, 데이터 관리 또한 사일로(silo) 방식에서 원본 데이터를 모두 모아서 통합하는 data lake 방식으로 변화하고 있다. 보안의 경우 전체 시스템을 통합하여 관리하던 firewall에 모든 패킷에 신원 정보를 추가하여 웹/모바일 등 다양한 소스의 통신에 신뢰성을 보장할 수 있는 zero trust 방식이 추가되어 발전하고 있다.

최근 미 국방부는 앞서 언급한 빠른 주기의 개발 및 운용에 대한 요구가 증가하고 있으며 안정적인 보안 제공의 필요성 또한 요구받고 있다. 이러한 배경으로 미 국방부는 DoD Enterprise DevSecOps Strategy Guide 문서를 필두로 DevSecOps Fundamentals, DevSecOps Reference Design, DevSecOps Playbook 등의 문서를 통해서 DevSecOps의 적용 방법을 구체화할 것을 홍보하고 있다[16,17].

미 국방부의 DevSecOps 기술은 웹 서비스부터 임베디드 시스템까지 모든 국방 관련 소프트웨어를 포함하며 산하의 여러 기관에서도 DevSecOps을 적용하는 방안을 수립하고 있다. 미 해군의 NAVSEA(Naval Sea Systems Command)에서 해양 무인로봇 담당 부서인 PM406의 Autonomy 가속화 개발 방안을 살펴보면 전체 개발 방법으로 DevSecOps를 적용하고 있음을 확인할 수 있다[18].

이와 같이 국방로봇의 소프트웨어 개발 분야에서도 DevSecOps 개발 방법이 적용되고 있으며 민간 분야의 소프트웨어 개발과 유사하게 무인화 관련 국방로봇의 경우 지속적인 요구사항 변경과 이에 대응하는 지속적인 소프트웨어 개발, 배포, 검증 등이 필요함을 알 수 있다.

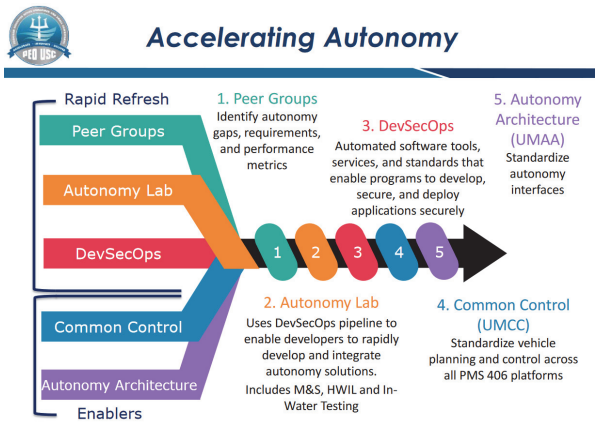

Fig. 5. DevSecOps in unmanned maritime autonomy architecture

3.2 MUM-T 기반 국방로봇 개발

최근 인공지능 기술 발전으로 인해 단순 원격 작업에 의해 움직이는 로봇에서 점차적으로 자율 판단이 들어가는 로봇 개발이 이루어지고 있다. 국방로봇 또한 인공지능 기술 적용을 통해서 운용자와 상호운용성을 높이는 방향으로 개발되고 있다. 하지만 어느 수준까지 국방로봇의 자율성을 부여하고 실제적인 작전 수행에 있어서 어떻게 운용하는 것이 효율적인가에 대해서는 아직까지 많은 연구가 필요한 실정이다.

MUM-T 는 인간과 로봇이 한 팀으로 작전을 수행하는 유·무인복합체계를 일컫는다. MUM-T를 위한 국방로봇은 운용자와 충분한 상호운용성을 가지도록 개발되지만 단독 판단의 수준에 한계를 갖도록 설계되고 최종적으로 공격과 같은 민감한 작전에 대해서 운용자가 철저히 통제하는 형태로 운용된다. 현재 미 해군이 계획 중인 UMS (unmanned maritime system)에서는 이와 같은 형태로 다양한 유·무인 복합 함정 및 세력들이 통합하여 운용하도록 개발되고 있다. 하지만 MUM-T 기반의 국방로봇들은 크게 다음과 같은 두 가지 문제점을 가지게 된다.

첫째, 요구사항의 불명확성이다. MUM-T 작전 운용에 있어서 무인체계의 요구사항을 명확하게 도출하는 것은 매우 힘든 일이다. 유인체계가 상호작용을 하면서 요구하는 능력이 작전에 따라 상이하고 무인체계 또한 유인체계의 특성이 계속 변화하는 형태를 가지기 때문에 이를 모두 만족시키는 요구사항을 한 번에 도출시키는 것은 매우 어렵다. 현재 미 해군 UMS의 경우는 순차적으로 함정을 건조하면서 요구사항을 지속적으로 업데이트하는 방식으로 무인체계를 개발하고 있다.

둘째, 만들어진 무인로봇의 성능 검증이 어렵다. 개발에 있어서 요구사항이 제시되었을 때, 각 요구사항에 대해서 시험과 검증 방법을 제시하고 성능을 검증할 수 있다. 하지만 요구사항이 불명확하고 MUM-T 간의 요구사항이 발생하기 때문에 하나의 요구 조건이 전혀 다른 조건에서도 시험이 되어야 하고 현실적으로 추가되는 시험 조건은 무한대로 볼 수도 있다[19].

따라서 MUM-T를 위한 국방로봇 개발을 위해서는 지속적으로 개발, 배포, 적용, 운용 및 피드백을 가지는 형태의 새로운 개발 방법이 필요하다.

3.3 DevSecOps 기반 지속 개발 모델

MUM-T를 위한 국방로봇은 실제 작전에 투입되는 것

이 최종 목표이다. MUM-T 작전을 수행하면서 국방로봇은 지속적인 운용(CO, continuous operations)이 가능해야 한다. 지속적 운용은 MUM-T의 작전에 최적화되는 형태로 국방로봇의 소프트웨어가 업데이트되어 사용되는 상태이다. 지속적 운용을 하기 위해서는 그 이전에 다양한 피드백 형태의 개발이 진행되어야 한다. Fig. 6는 DevSecOps의 개발 수명주기와 이에 해당하는 지속 개발의 주요 개념들을 나타낸다.

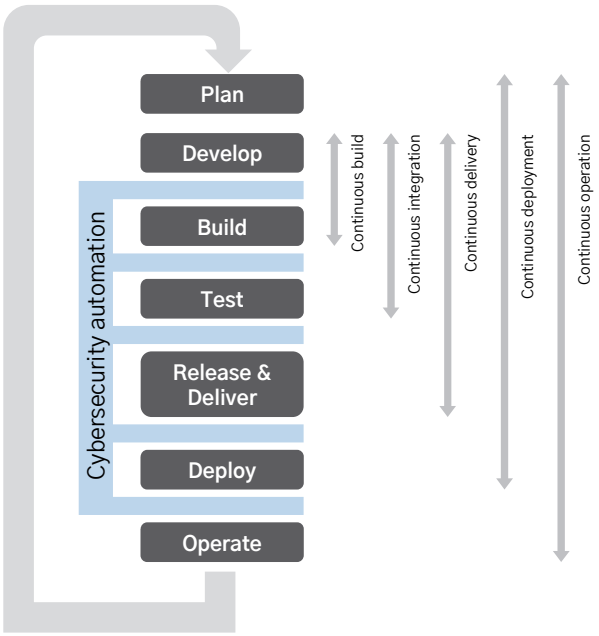

Fig. 6. Unfolded DevSecOps lifecycle

가장 작은 단계로 지속 개발(CB, continuous build)은 코드 개발과 빌드 사이의 피드백이다. 요구사항 변경에 따라 계획이 수립되고 코딩이 수행되며 자동 빌드가 진행된다. CB 단계에서 보안 점검을 위해서 자동화된 보안 프로토콜이 적용되며 빌드 또한 자동으로 수행된다. 지속 통합(CI)은 개발 CB가 완료된 코드에 대해서 테스트를 통해서 통합 절차를 이룬다. 테스트 또한 자동으로 이루어지도록 설정이 되며 빌드된 코드 또한 보안 검사를 통해서 안정성을 확인받는다. 이때 테스트는 단순히 개발된 코드가 아닌 전체 시스템에 대한 테스트로 진행된다. 테스트에서 가장 중요한 부분은 자동화이다. 자동화를 통해서 기존에 관련되는 모든 테스트 항목과 신규 요구사항에 의해서 발생한 테스트 항목을 종합하여 테스트를 수행하여야 한다.

지속 배포, 적용(CD, continuous delivery & continuous deployment)은 테스트 완료된 코드를 배포하고 적용하는 단계이다. 코드 릴리즈와 배포는 시스템 구성 단계에 따라서 간소화될 수도 있다. 개발 장소와 적용 장소가 다를 경우 또한 다양한 네트워크를 통해서 테스트된 코드를 전달되는 과정에서 오염이 될 수도 있다. 따라서 모든 릴리스, 배포 과정에서 보안 검사를 수행하도록 되어 있다.

DevSecOps 의 지속 개발 모델은 다양한 자동화 도구를 기반으로 한다. 또한 배포되고 운용되는 소프트웨어를 단위 실행 파일이나 라이브러리가 아닌 컨테이너 개념에서 관리를 하고 배포를 수행한다. Fig. 7은 미 국방부에서 예제로 설명하고 있는 DevSecOps 적용을 위한 다양한 애플리케이션을 나타낸다. 그림에서 나타난 다양한 애플리케이션은 대부분 DevOps에서 검증된 애플리케이션으로, 기본적으로 Jenkins를 기반으로 CI를 수행하며 Gitlab을 이용한 소스관리와 kubernetes를 이용한 배포를 수행하는 형태를 가진다. 각각의 계획, 코딩, 빌드, 테스팅, 배포 단계별로 추가적인 애플리케이션을 선택하고 최종적으로 오케스트레이션을 통해서 전체 개발 환경을 구축해야 한다.

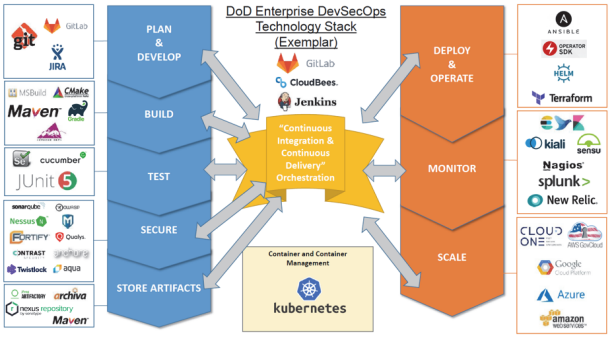

Fig. 7. Various applications for DevSecOps execution[16]

3.4 국내 적용 방안

서론에서 언급한 바와 같이 국방로봇 소프트웨어는 무기체계 소프트웨어 개발 지침을 기본으로 다양한 국내 개발 규정을 준수하면서 개발되어야 한다. DevSecOps와 국내 소프트웨어 개발 지침과의 가장 큰 차이점은 기존의 소프트웨어 개발 지침이 폭포수 모델을 기반으로 한다는 점이다. 따라서 DevSecOps를 적용하기 위해서는 전체적인 개발 절차를 수정해야만 한다. 하지만 국방 분야의 선 적용 사례가 있기 때문에 적용에 있어서 시행착오를 줄일 수 있을 것으로 판단된다. 다만 DevSecOps 방식은 단순히 개발 방법론만이 아닌 개발을 대하는 문화를

같이 바꾸어야 정착이 된다. 현재와 같이 개발 후 한 번의 시험 평가를 통해서 사업의 성패가 판단되는 형태의 문화에서는 DevSecOps를 적용할 수 없다. 최근 미 해군에서 발표된 Intelligence Autonomous System 개발 계획에서도 새로운 무인체계 도입의 실패가 실패를 두려워하는 문화라고 지적하고 있음을 생각해 볼만 한다.

4. 결론

MUM-T 기반 국방로봇은 기존의 무기체계와 다른 형태로 개발이 수행되는 신개념의 무기체계이다. 국방로봇은 요구사항을 정하기 힘들고, 운용시 추가적인 요구사항들이 나오며 테스트 기간이 오래 걸리는 도전적인 문제를 가지고 있다. 특히 국방로봇은 소프트웨어 개발이 중요하며 기존의 무기체계에서 사용하는 개발 방법으로는 개발의 한계를 가진다.

미 국방성에서 새로운 무인체계 개발에 적용하고 있는 DevSecOps 기반 개발 방법은 다양한 국방 로봇 소프트웨어 개발 문제를 해결할 수 있는 좋은 대안이다. 하지만 DevSecOps 개발 방법을 국내 적용하기 위해서는 기존의 무기체계 소프트웨어 개발 지침을 수정해야 하며 시험평가가 일회성이 아닌 지속적인 시험평가 방식으로 개발 문화가 바뀌어야 한다.

최근 많은 형태의 무인체계 개발 계획이 수립되고 있다. 무인체계의 개발이 단순히 하드웨어 개발이 아니라 소프트웨어 개발이 중요하다는 점을 인지하고 앞으로 개발될 다양한 무인체계를 효과적으로 개발할 수 있는 방법을 고민할 시간이다.

참고문헌

[1] A. Makarenko, A. Brooks, and T. Kaupp, "Orca: Components for robotics," in International Conference on Intelligent Robots and Systems (IROS). Citeseer, 2006, pp. 163-168.
[2] L. B. R. Oliveira, F. S. Osório, and E. Y. Nakagawa, "An investigation into the development of service-oriented robotic systems," in Proceedings of the 28th annual ACM symposium on applied computing, 2013, pp. 223-228.
[3] A. Elkady and T. Sobh, "Robotics middleware: A comprehensive literature survey and attribute-based bibliography," Journal of Robotics, Vol. 2012, 2012.
[4] Weapon system software development and management manual, DAPA.
[5] Defense Architecture Framework, DAPA.
[6] Defense CBD Method, DAPA
[7] https://www.navsea.navy.mil/Home/USC/Program-Offices/PMS406/PMS406
[8] B. Bauml and G. Hirzinger, "Agile robot development (ard): A pragmatic approach to robotic software," in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2006, pp. 3741-3748.
[9] D. Brugali and A. Shakhimardanov, "Component-based robotic engineering (part ii)," IEEE Robotics & Automation Magazine, Vol. 17, No. 1, pp. 100-112, 2010.
[10] I. Bucena and M. Kirikova, "Simplifying the devops adoption process," in BIR Workshops, 2017.
[11] L. E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä, J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo, and C. Lassenius, "Devops in practice: A multiple case study of five companies," Information and Software Technology, Vol. 114, pp. 217-230, 2019.
[13] R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, "What is devops? a systematic mapping study on definitions and practices," in Proceedings of the Scientific Workshop Proceedings of XP2016, 2016, pp. 1-11.
[14] Dr. Robert Grabowski, Big Picture for Autonomy Research in DoD, Soft and Secure Systems and Software Symposium, Jun 2015.
[15] The Role of Autonomy in DoD Systems, Defense Science Board, 2012.
[16] DoD Enterprise DevSecOps Fundamentals, Ver 2.0, May, 2021.
[17] DoD Enterprise DevSecOps Strategy Guide, Ver 2.0, March, 2021.
[18] Unmanned Maritime Autonomy Architecture (UMAA), PM406, 2020.
[19] Heather M. Wojton, Test & Evaluation of AI-enabled and Autonomous Systems: A Literature Review, Institute for defense analyses white paper, 2020.