정 보 보 호 학 회 지
제25권 제1호, 2015. 2

보안을 고려한 무중단 환경에서 개발운영조직 통합관리(DevOps)

전 인 석*

요 약

보안에서 인적보안(Personal Security)의 예방통제 중 하나로 개발/운영을 분리하는 직무분리(Separation of Duty)를 해오고 있다. 고의적인 시스템의 오용을 줄이기 위한 방법이며, 많은 국제 표준과 국내 표준(COBIT, ISMS, 등)에서 직무분리를 명시하고 있다. 하지만 무중단 시스템이면서, 운영자가 특정 전문가 집단으로 한정되고, 수 많은 변경이 발생하는 무중단 환경에서 개발/운영이 분리 됨으로 인한 여러 가지 문제점이 발생 하고 있다. 체계를 운영하면서 전문지식을 기반으로 한 요구사항을 명확하게 이해하지 못하면, 추가적인 요구사항이 발생한다. 이는 체계의 품질저하와 위험(Risk)증가로 이어지게 된다. 따라서 본 연구에서는 개발운영조직 통합관리(DevOps) 방법론을 SCADA와 같이 운영자가 해당 분야의 전문성을 가지고, 무중단으로 운영되며, 수많은 변경이 반영되는 시스템에서 개발 및 운영을 통합하였을 때, 발생할 수 있는 문제점과 개선방안을 제시하고자 한다.

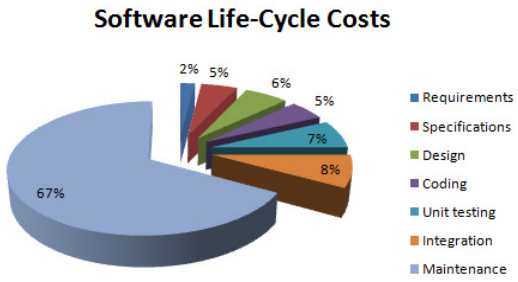
Ⅰ. 서 론

소프트웨어에서 사용되는 비용 중에 유지보수에 들어가는 비용이 50%가 넘어가고 있다. 비용뿐만 아니라 자원의 효율성과 운영환경의 연속성을 위해서 개선을 하고자 하는 노력은 계속되어 왔다.

대규모의 폭포수 개발방식 프로젝트들이 점진적 배포와 나선형 개발방식으로 바뀌고 있고, 개발과 운영을 통합하는 개발운영조직 통합관리에 이르게 되었다. 이는 개발과 운영을 통합하여 운영의 효율성을 극대화 하는 것이다.

이는 대형 인터넷기업과 무중단 환경에서 특정 전문가 집단이 사용하는 환경에서 그 필요성이 더 크다고 볼 수 있다. 무중단 환경에서는 서비스가 유지되고 있는 상태에서 배포를 해야하고, 서비스 장애가 발생하면 안되기 때문에 운영자와 개발자간의 커뮤니케이션이 매우 중요하다.

이와 같은 환경에서 개발과 운영을 통합하였을 때, 생길 수 있는 문제점과 개선방향에 대하여 제시하고자 한다.



〔그림 1〕 소프트웨어 Life-Cycle 비용[9]

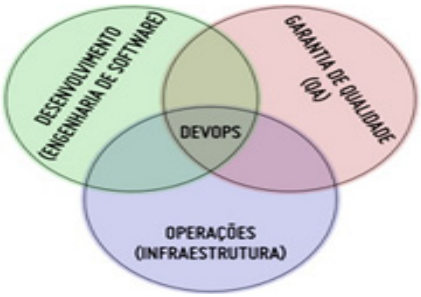
Ⅱ. 개발운영조직 통합관리(DevOps)

2.1. 개발운영조직 통합관리(DevOps)의 필요성

개발(Development)과 운영(Operations)의 합성어로서 소프트웨어 개발자와 정보기술 전문가 간의 소통, 협업 및 통합을 강조하는 개발방법론 이다. 개발운영조직 통합관리(DevOps)는 소프트웨어 개발조직과 운영조직 간의 상호 의존적 대응이며 조직이 소프트웨어 제품과

본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터육성지원사업/IT융합고급인력과정지원사업의 연구결과로 수행되었음 (NIPA-2014-H0301-14-1023)

* Ahnlab CERT팀 전인석(wilcois@ahnlab.com)



〔그림 2〕 개발운영조직통합관리의 개념

서비스를 빠른 시간에 개발 및 배포하는 것을 목적으로 한다.[1]

개발운영조직 통합관리(DevOps)는 2012년부터 급속하게 많은 분야에서 연구되고 있으며, 본 논문에서는 무중단 환경에서의 적용에 대하여 연구하였다. 일반 SW 환경에서도 개발/운영 통합으로 많은 효과가 있지만, 특히 무중단으로 운영되는 시스템에서 전문적 지식을 가지고 한정된 운영자가 운영하는 환경에서는 그 효과가 배가 된다. 전문지식이 프로세스에 녹아 있고, 운영자가 개발 초기의 요구사항 단계부터 개발의 모든 단계에서 참여하지 않는다면 최초의 목적을 달성하기 어렵고 이는 SW개발 비용의 증가와 장애로 이어질 수 있기 때문이다.[2]

일반적인 IT제품의 경우는 기획자가 제품을 기획하고 개발을 하고, 기획자에 의해 제품이 계속해서 업데이트

〔표 1〕 IT와 OT의 차이

	IT	OT
목적	트랜잭션 처리,정보제공	장비제어 및 모니터링
아키텍처	범용 어플리케이션	실시간,이벤트-드리븐
운영자	일반 사용자	운영자, 공학자, 기술자
연결	회사 네트워크, IP기반	통제 네트워크 IP기반
요구사항	SW기능 범위 내 일반적인 요구사항	전문적이고 구체적인 요구사항
가용성	지연이나 중단이 용납됨	24X7X365(지속성)
패치빈도	정기적/예정됨	비정기적/빈도가 높음
QA	별도 QA팀에서 진행	개발에서 진행

가 이루어 진다. 보안 취약점에 의해 일부 긴급 업데이트가 이루어지기도 하지만, 대부분의 경우에는 사전에 계획된 순서와 일정으로 업데이트가 이루어지게 된다.

하지만 OT(본 연구에서는 무중단 환경)의 경우는 기획자가 초기에 계획을 하고 개발을 하는 것 까지는 동일하지만 IT환경에서처럼 계획적인 업데이트를 할 수가 없다. 대부분의 경우 전문가 집단으로 이루어진 운영자가 운영을 하면서 제어 시스템을 개선하고 외부 또는 내부의 환경 변화에 따라서 요구사항이 도출되기 때문이다. 무중단 모니터링 시스템에서 모니터링 해야하는 자산 또는 지점이 증가하였을 경우 서비스 중단없이 모니터링 기능이 추가되어야 하며, 개별 자산의 특징이나 특성이 반영 되어야 한다. IT환경에서의 보안 긴급업데이트와는 성격이 다르며 기존 시스템과 유기적으로 연동되고 상호작용하는 OT환경에서는 충분한 QA를 거치지 않고 반영되는 경우가 있을 수 밖에 없다.

개발과 운영은 라이프사이클 속에서 상호 보완적으로 연결이 되어야 하지만, 업무와 책임자가 분리되어 있어, 상호 협업에 난관이 있다. 이는 운영자가 전문가 집단으로 한정되어 있는 환경에서 매우 두드러지게 나타난다. 특히, 클라우드 컴퓨팅 환경에서는 하드웨어에서 제어하는 영역이 소프트웨어로 넘어가면서 개발과 운영에 더 많은 협업이 필요하게 되었다. [4] 대부분의 개발상의 오류나, 요구사항 재요청은 이런 개발과 운영의 원할한 커뮤니케이션의 부족에서 발생하고 있다.

개발운영조직 통합관리(DevOps)의 가장 큰 장점은 개발과 운영의 원활한 의사소통과 협업으로 인하여, 적시에 추가가능을 제공하고, 운영자의 요구사항이 정확하게 반영되어 업무 효율성이 증가되는 선순환 구조를 이룰 수 있는 것이다. 그로 인해서 넷플릭스, 페이스북, 아마존, 트위터, 구글 등 선진 기업들에서 폭넓게 채택하고 있다.[5] 서비스의 규모가 매우 크기 때문에 계속해서 개발이 반영되는 환경이거나, 실시간으로 모니터링하는 환경에서는 개발운영조직 통합관리(DevOps)의

〔표 2〕 개발과 운영의 환경 차이

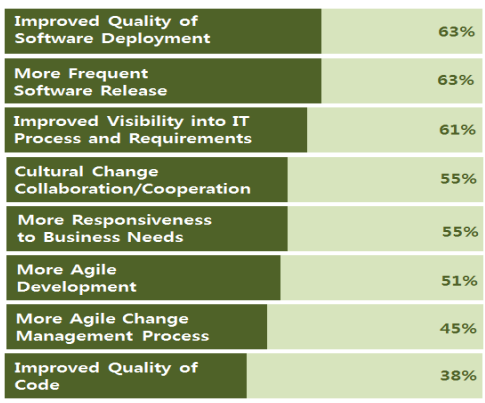
	개발	운영
목표	기능추가, 릴리즈 배포,	가용성, 장애감소
성향	변화추구	안전하게 유지
성과척도	새로운 기능을 추가한 수와 기간	서버의 가동시간 및 무중단기간

필요성을 느끼고 도입이 증가되고 있다.

2.2. 개발운영조직 통합관리(DevOps)의 효과성

가트너가 발표한 2015년도 10대 전략 기술 동향을 보면 2014년에 이어서 웹-스케일 IT가 있다. 웹-스케일 IT는 즉각적으로 실현되진 않겠지만 상업적 하드웨어 플랫폼이 새로운 모델을 수용하고 '클라우드 최적화', '소프트웨어 정의'에 대한 접근들이 주류화되면서 점차 발전할 것이다. 다수의 기업이 웹 스케일 IT를 실현할 수 있는 첫 걸음은 개발운영조직 통합관리 (DevOps)이다. 개발운영조직 통합관리를 통해 애플리케이션과 서비스의 빠르고 지속적인 점진적 개발이 가능하며, 개발과 운영을 동시에 진행할 수 있다.[7]

개발운영조직 통합관리(DevOps)를 성공적으로 조직에 적용한 사례를 보면 다음과 같은 효과가 나타났다. Puppet Lab에서 발표한 2013 State of DevOps Report에 따르면, 4000명 이상의 IT 운영 전문가들을 대상으로 설문조사를 진행한 결과 IT기업의 규모와 상관없이 개발운영조직 통합관리(DevOps)를 적용하는 사례가 급증하고 있다는 흐름이 나타났다. 응답자의 63%가 개발운영조직 통합관리(DevOps) 프랙티스를 수행했다고 응답했고, 이는 2011년 이후 26% 증가한 수치이다. 개발운영조직 통합관리(DevOps) 적용에 성공한 조직의 응답자들은 소프트웨어 배치의 품질이 향상되었고 소프트웨어 배포 횟수가 상당히 많이 증가했다고 응답했다. 이러한 결과는 개발운영조직 통합관리(DevOps)의 사업적 가치까지 입증된 것으로 볼 수 있다. 개발운영조직 통합관리(DevOps)를 적용함으로써 민첩성(agility)과 신뢰성(reliability)을 기반으로 한 체계의 고성능(High Performance)을 보장할 수 있게 된 것이다.



〔그림 3〕 개발운영통합관리의 효과

이 외에도 IT 프로세스와 요구사항에 대한 가시성도 61% 향상되었다고 한다. 이는 요구분석 단계부터 개발자와 운영자가 함께 참여함으로써 나타난 효과라고 분석할 수 있다.[2]

Ⅲ. 개발운영조직 통합관리(DevOps)로 인한 문제

3.1. 통합으로 인한 변화

소프트웨어의 개발 방법은 전통적인 폭포수 개발방식에서 점진적 배포와 나선형 개발방식으로 변화하고 있다. 이는 개발이 완료되고 배포되는 것이 아니라 배포되고 운영하고 있는 상태에서 개발이 이루어지고 배포가 발생한 다는 것을 의미한다. 충분한 테스트를 하지 못하고 배포가 될 수 있음을 의미한다. 따라서 운영중에 버그나 개선사항이 많이 발생을 하며, 이는 다시 개발로 전달되어 바로 조치가 이루어 진다. 개발에서 발생할 수 있는 리스크가 더욱 커지게 된다.

개발과 운영이 통합되어 관리되면서 개발자가 사전에 일정관리를 하기가 힘들어 진다. 이는 계획된 일정의 지연이나 업무우선순위의 변화를 발생시키며, 업무효율성이 떨어 질 수 있다. 개발자의 생산성만을 놓고 본다면 효율성이 감소가 되지만, 전체 비즈니스 측면에서 개발운영조직통합관리가 효율성이 좋다.

개발운영조직에서는 개발자의 업무우선순위가 수시로 변화하기 때문에 이를 관리할 필요가 있다. 운영에서 수시로 발생되는 요구사항의 우선순위를 명확하게 정의하고, 우선순위에 따라 업무를 처리해야 한다. 수시로 운영자의 요구로 인해 개발자의 업무가 중단이 되겠지만, 이는 개발운영조직에서는 당연한 결과이다. 개발운영조직에서의 개발자는 개발자이면서 운영자이기도 하기 때문에, 빠르게 개발이슈를 해결하고 운영에 투입되어야 한다.

개발운영조직에서는 개발자가 일하는 방식이 전통적인 방식이 아니기 때문에 평가나 목표가 변화되어야 한다. 전통적인 기능의 수나 코드의 라인 수로 개발자의 생산성을 평가하는 것은 개발운영조직에서 적당하지 않다.

개발운영조직에서는 과거 모델에서 개발에 비중이 높았다면, 운영에 비중이 더 높은 것이다. 운영중에 요

구사항을 반영하면서 개발자의 대부분의 리소스는 유지보수 업무에 투입이 된다. 무중단 환경의 운영자는 모니터링 시스템을 개선하고 설계할 수 있다. 전통적인 모델에서 개발자가 고민하던 영역을 운영자가 고민을 하는 것이다. 운영자의 요구사항은 매우 구체적이고 정량적이며, 전문적이다.

운영자는 인프라 설계나 코드의 재사용성을 고려한 요구사항 등을 하게되고, 개발자는 평소에 운영을 하거나 개발을 하다가 요구사항이 발생하면 운영의 관점에서 유지보수를 진행하게 된다. 이와 같이 개발운영조직을 통합하게 되면 개발과 운영의 경계가 매우 낮아지게 된다.[8]

3.2. 통합으로 인한 보안 문제

직무분리는 예방통제의 하나로서 그 중에 개발/운영의 분리는 기본으로서 거의 모든 IT지침에서 명시하고 있다. 개발자가 운영자 환경에 접근가능한 경우 발생할 수 있는 문제점과 반대로 운영자가 개발자 환경에 접근했을 때 발생할 수 있는 문제점을 구분해서 분석하고 해결 할 필요가 있다.

개발자가 운영시스템에 접근이 가능할 경우 개발코드가 운영시스템에 적용 될 경우 서비스에 매우 큰 영향을 끼치게 되거나 TEST코드에 의해 사용자에게 보이지 말아야 할 부분이 노출되기도 한다. 개발운영조직 통합관리(DevOps)를 도입하더라도 개발시스템과 운영시스템은 물리적으로 구분되어 하고, 배포되지 않은 코드가 운영시스템에 반영되지 않도록 하는 통제가 필요하다.

내부부정이 발생할 수 있는 ERP시스템과 같은 부분에서 운영자가 개발시스템에 접근이 가능할 경우 소스코드의 구성을 알게 되면서 내부부정이 발생할 수 있다. 무중단 환경에서 개발에 투입되는 요구사항은 일반 환경과 다르다. 운영자는 소수의 전문가 집단으로 한정되어 있고 전문성을 가지고 업무를 진행을 하게 된다. 실제로 대부분의 요구사항은 운영자가 정교하게 정량적인 수치나 함수로 이루어진 기능이거나 시스템 오류를 탐지하기 위한 스크립트나 정규표현식이 대부분 이다. 일반 사용자에게 배포하는 시스템과 다르게 요구사항을 운영자가 전달하고 개발자가 개발을 완료하면 다시 운영자가 사용을 하는 환경에서는 개발에서 충분한 테스트를 진행한다 하더라도 특수한 환경에서 사용하고 사용환경 자체가 빠른 속도로 변화하기 때문에 정확한 테스트를 진행하기 어렵다. 요구사항이 정교하고 복잡하기 때문에 직접 개발을 하지 않더라도 시스템의 구현과정과 내부로직을 알아야만 정확한 요구사항이 전달되고 검증될 수 있다는 것이다. 즉, 운영자가 개발에 초기단계부터 검증단계 까지 접근하는 개발운영조직 통합관리(DevOps)를 함으로서 정확한 요구사항 분석으로 추가 요구사항을 발생을 억제시키고, 그것으로 인하여 체계의 품질향상을 가져온다. 부정확한 요구사항 분석은 내부부정만큼의 위험요소이다.

〔표 3〕 표준에서의 직무분리

표준	지침
ISMS	6.1.2 직무별 권한과 책임을 분산시켜 직무 간 상호견제를 할 수 있도록 직무 분리 기준을 수립하여야 한다. - 개발과 운영 직무 분리 (필수) - 정보시스템(서버, DB, 네트워크 등)간 운영 직무 분리 - 정보보호 관리와 정보시스템 운영직무 분리 - 정보보호 관리와 정보시스템 개발직무 분리 등
ISO 27799	6.1.2 조직의 자산에 대해서 의도하지 않거나 권한 없는 수정 또는 오용하는 경우를 줄이기 위해서 직무나 책임은 분리돼야 한다.
COBIT	10.1.3 모든 정보처리설비의 관리와 운영을 위한 책임과 절차가 수립되어야 한다. 이는 적절한 운영절차를 수립하는 것을 포함한다. 적절한 경우, 부주의하거나 고의적인 시스템 오용 위험을 줄이기 위한 직무 분리를 이행하여야 한다.

개발자가 운영시스템에 접근할 수 있는 문제는 물리적으로 분리하여 기존과 동일하게 예방해야 하고, 내부부정의 문제는 예방통제가 아닌 탐지통제와 감사로서 위험을 감소시켜야 한다. 이는 개발운영조직 통합관리(DevOps)를 도입함으로서 얻는 효율성이 내부부정이 발생할 위험보다 클 경우에 도입이 가능하다. 대부분의 전문가 시스템의 운영자 요구사항의 전문성과 복잡성을 고려할 때, 실익이 더 크다고 볼 수 있다.

Ⅳ. 개발운영조직통합관리(DevOps)의 개선방향

- 충분한 테스트를 하지 못하는 문제

무중단환경의 개발운영조직은 프로젝트로서 거대한

프로그램을 만들어 내지 않는다. 빠르게 개발하여 서비스를 하고 해당 서비스를 단절 없이 유지하는 것이 목표이다. 무중단환경의 특성상 충분한 테스트를 거치지 않은 업데이트로 인한 서비스단절이 생겨서는 안된다. 따라서, 개발운영조직통합의 신속함은 유지하면서 신뢰성을 높이기 위하여, 표준화하고 자동화 하는 것이 필요하다. 가장 중요한 것은 운영에서 개발로 이어지는 선순환의 피드백이 이루어질 수 있는 프로세스가 정립되어야 한다.

- 개발자의 성과를 측정할 지표

전통적 환경에서의 개발자와 개발운영조직에서의 개발자의 성과지표는 같을 수가 없다. 기존의 얼마나 빠르게 많은 기능을 개발 하였는가가 지표의 핵심이라면, 개발운영조직 환경에서는 서비스가 얼마나 중단 없이 운영이 되었는지, 개발된 기능의 사용빈도가 높은지가 척도가 된다. 즉, 운영의 성과지표를 개발자도 함께 가지고 가는 것이다. 이는 개발과 운영 모두의 궁극적인 목표인 서비스의 향상과 신뢰성 증대를 위하여 조직이 통합되었기 때문에 결과를 성과지표로서 가져가야 한다.

- 개발운영조직 통합에 의한 보안

개발운영조직을 통합하더라도 개발과 운영시스템은 물리적으로 분리되어 있어야 한다. 운영자의 전문적인 요구사항에 명확하게 대응하기 위하여 개발운영조직을 통합하였으나, 기존의 IT표준에서 말하는 직무분리의 필요성까지 없어지는 것은 아니다. 개발과 운영의 통합으로 인한 내부비리는 탐지통제와 감사로서 리스크를 감소시키고, 그 외의 물리적인 통제는 유지해야 한다. 이는 의도치 않은 실수에 의한 사고를 막기 위함이다. 개발운영조직 통합의 가장 긍정적인 효과는 불필요한 절차를 줄이고 개발자와 운영자간의 커뮤니케이션이다. 따라서 사람과 조직은 통합하더라도 시스템은 분리하여야 하며, 개발자나 운영자의 실수로서 발생할 수 있는 부분은 자동화 시스템과 Tool로서 감소 시켜야 할 것이다.

〔표 4〕 개발운영조직 통합환경에서의 개발자 성과지표

구분	성과지표
일반적	라인수 기능 점수 특성 점수 스토리 점수 개발 속도
개발운영	운영장애 빈도 기능의 사용도 무중단 운영시간 요구사항 접수 후 업데이트 시간 유지보수에 의한 업데이트 수

Ⅴ. 결 론

개발운영조직 통합관리는 하나의 흐름으로서 확산되고 있다. 개발과 운영의 효율성을 극대화 시키는 방법이지만, 보안에서 직무분리의 기본이기도 하다. 개발운영조직 통합관리가 처음에 기대했던 효과를 이루기 위해서는 보안적 통제나 프로세스보다도 기업 문화의 변화가 가장 중요하다.

개발운영조직 통합관리는 앞으로 더 많은 조직에서 적용하게 될 것이고, 문제점들을 개선하면서 정착 될 것이다.

참 고 문 헌

[1] Seonghyun Yun, "A study on international standards and safety requirements for the development of automotive safety-related software", KSAE, 2009.

[2] 이은정, 개발운영조직 통합관리(DevOps) 프로세스 연구, 고려대학교 융합소프트웨어전문대학원 석사논문 2015

[3] Mike Loukides, 이덕준, What is DevOps?, 한빛네트워크, 2012

[4] Bill Phifer, "Next-Generation Process Integration : CMMI and ITIL Do Devops", Cutter IT Journal, p.28~p.33, August 2011

[5] 강송희, "한눈에 보는 실전 클라우드 프로젝트", 에이콘, 2013년 11월

[6] Michael Httermann, "DevOps for Developers", Apress, 2012

[7] http://www.gartner.com/newsroom/id/2867917

[8] http://java.dzone.com/articles/devops-isnt-killing-developers

[9] http://www.diag.com/

〈저 자 소 개〉

전 인 석 (In-seok Jeon)
2009년 8월 : 건국대학교 정보통신대학원 정보보호학과 석사
2009년 9월~현재 : Ahnlab CERT 팀 주임 연구원
관심분야 : 네트워크보안, ISMS, DevOps