

데브옵스(DevOps) 개발 프로세스 모델

2021년 11월 18일
충북대학교 대학원
산업인공지능학과

차 례

요약	3
Ⅰ. 서 론	3
1. 데브옵스 정의	
2. 탄생 배경	
Ⅱ. 본 론	4
1. 데브옵스의 특징	
2. 데브옵스의 방법	
3. 데브옵스 툴체인	
4. 애자일과의 차이점	
Ⅲ. 본 론	6
참고 문헌	7

요 약

데브옵스(Devops)는 개발(Development)과 운영(Operation)이 합쳐진 단어지만 각각의 용어를 결합한 이상의 포괄적인 아이디어와 방식을 나타낸다.¹⁾ 과거의 개발 방법론이 담지 못하는 소프트웨어 제품 전반의 계획과 개발 운영에 대한 내용을 담고 있다. 그리고 끊임없는 개발과 배포를 통해 고객의 눈높이를 만족시키려는 개발 모델이다. 이 보고서에서는 데브옵스에 대한 정의와 특징 등에 대한 내용을 기술할 것이다.

I. 서 론

1. 데브옵스 정의

데브옵스는 소프트웨어의 개발(Development)과 운영(Operation)의 합성어로서 소프트웨어 개발자와 정보 기술 전문가 간의 소통, 협업 및 통합을 강조하는 개발 환경이나 문화를 말한다. 데브옵스는 소프트웨어 개발 조직과 운영 조직 간의 상호 의존적 대응이며 조직이 소프트웨어 제품과 서비스를 빠른 시간에 개발 배포하는 것을 목적으로 한다.²⁾ 즉 소프트웨어 개발 및 배포 주기를 단축시키기 위한 개발 모델이라 할 수 있다.

2. 탄생 배경

1) 비즈니스 환경의 변화

온라인 환경에서 모바일 환경으로 넘어오면서 기업 간의 경쟁이 온라인 강자 위주(MS, Yahoo)의 시장에서 절대 강자가 없는 시장으로 변화하고 있다. 시장 상황이 기존 온라인 강자들과 더불어 신규 업체들도 모두 모바일 시장 혹은 하이브리드 형태의 서비스 시장에 투자하고 있다. 이에 따라 다른 업체들과 차별화된 서비스로 고객을 잡기 위해 고객의 피드백에 대한 빠른 개선이 필요하게 되었다. 이처럼 비즈니스 민첩성 향상을 통해 업무 혁신의 가속화가 필요한 환경이 되었다.

2) 지속적인 소프트웨어 배포의 필요성

비즈니스 환경의 변화에 따라 지속적인 서비스의 개선을 통한 소프트웨어 배포가 필요하게 되었다. 그러나 기존의 개발 조직과 운영 조직 그리고 QA 조직 사이의 작업/소통 문화의 차이와 각 조직의 업무에 대한 관점의 차이로 지속적이고 빠른 서비스 배포에 어려움이 증대되었다. 또한 지속적인 서비스의 배포를 위해서는 소프트웨어의 통합과 테스트의 속도도 그에 맞춰 진행되어야 한다.

3) 안정적인 운영의 필요성

단순하게 지속적이고 빠른 서비스의 배포가 아닌 운영의 안정성 또한 요구되었다.

4) Cloud의 등장과 Application Architecture의 변화

Cloud 서비스 등장하기 전에는 Sever 및 Database는 운영의 영역이었다. 그러나 Cloud의 등장으로 Server, Database 관리의 외주화가 가능하게 되었으며 그로 인한 관리 포인트(보안, 관제 등)가 줄게 되었다. 그리고 Application

Architecture가 기존의 Monolithic 구조에서 가상화 플랫폼(Docker, Kubernetes 등)을 이용한 Micro Service 형태로 모듈화가 진행되어 기존보다 손쉬운 배포가 가능해지고 유지보수 또한 편하게 되었다.

II. 본 론

1. 데브옵스 특징

1) 빠른 작업 속도와 배포

개발부터 배포까지 기존의 모델대비 빠른 작업 속도와 배포 속도를 가진다. 그리고 이로 인해 클라이언트에 빠른 대응이 가능하게 되었다.

2) 협업으로 인한 업무 효율 증대

개발팀과 운영팀, 테스트팀 간의 긴밀한 협업이 필요하기 때문에 팀 간의 비효율성을 개선하는 목표의식 가지게 된다. 데브옵스 도입(개발 / 운영 Tool, 기업문화의 변화 등)으로 운영과 개발의 업무의 간극이 줄어 프로젝트에 따라 같은 팀에 개발과 운영을 두는 경우도 생기게 된다. 이에 따라 최소 인원으로 개발과 운영을 두는 경우도 생기게 되어 서비스와 개발의 운영 합리화로 비용의 절감이 가능해 진다.



그림 1) 개발과 운영의 간극³⁾

2. 데브옵스 방법

조직에서 개발 및 제품 출시 속도와 향상을 위해 일반적으로 사용되는 데브옵스 방법은 스크럼(Scrum), 칸반(Kanban), 애자일(Agile)을 들 수 있다.

1) 스크럼

스크럼은 개발 및 QA, 배포를 위한 프로젝트를 가속하기 위한 팀원의 협력방법을 의미한다. 스크럼은 Agile 방법론에서 사용하며 작은 주기(Sprint)로 개발 및 검토를 하며 효율적인 협업 방법을 제시한다. 비즈니스 요구를 충족시키기 위해, 작은 목표를 짧은 주기로 점진적이며 경험적으로 제품을 지속적으로 개발하는 관리 프레임워크를 이야기 한다.⁴⁾

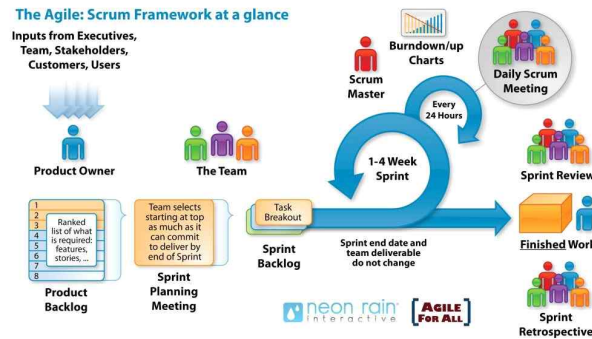


그림 2) 스크럼⁵⁾

2) 칸반

칸반이란 칠판에 진행상태나 사람, 업무 종류, 업무 중요도 등에 따라 칸을 나누고 각 칸에 이슈 내용이 적힌 메모지를 붙여 현재 업무 현황을 한눈에 파악하는 시스템이다. 애자일 개발 프로세스 전반에 걸친 적시 개발을 지원하는 방법론이자 시스템이다. 대표적인 툴로는 Asana, Jira등을 들 수 있다.

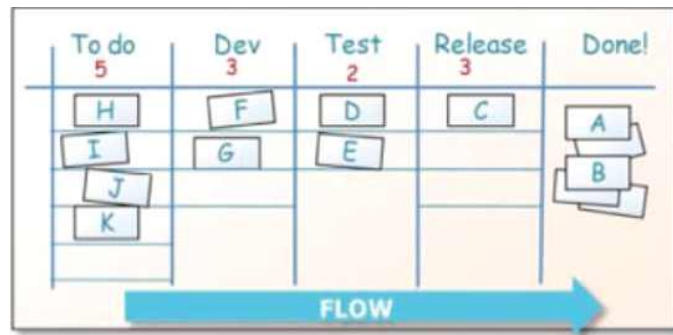


그림 3) 칸반⁶⁾

3) 애자일

애자일 소프트웨어 개발 방법론은 데브옵스에 많은 영향을 미치고 있다. 애자일은 기존의 폭포수 모델처럼 긴 개발 방법 대신 짧은 소프트웨어 개발 라이프 사이클을 사용할 것을 지시한다. 애자일은 고객의 요구 사항이 변화하는 현실에서 생산성을 유지하고 빠른 프로그램 릴리즈를 추진하도록 설계된 개발 방법론이다. 또한 데브옵스는 애자일의 연장선이라 볼 수 있다.



3. 데브옵스 툴체인

개발 및 서비스 환경의 변화가 개발 지향에서 서비스 지향 형태로 변화하게 되었다. 그로 인해 빠른 서비스의 릴리즈를 위한 다양한 자동화 툴과 협업툴의 필요성이 나타나게 되었다. 이런 툴들의 집합을 툴체인(toolchain)이라 한다. 데브옵스 툴체인으로 많이 사용되는 도구로는 Jira, SVN, GitHub, Jenkins, Docker, Kubernetes 등이 있다.

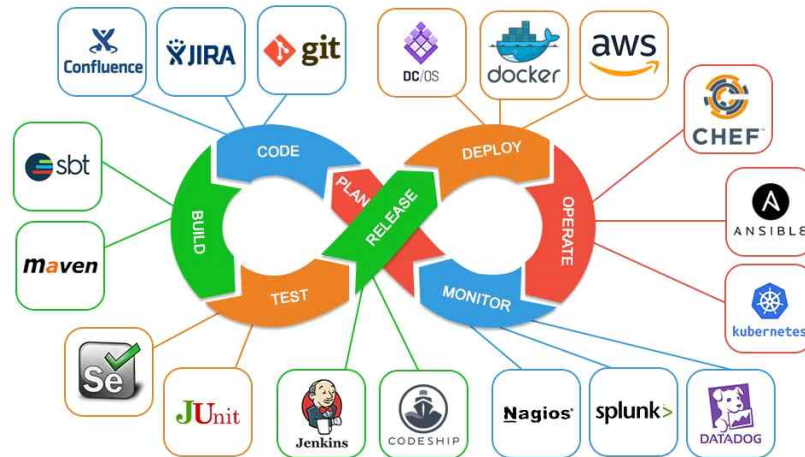


그림 5) DevOps toolchain8)

그림 5)처럼 데브옵스 진행 단계는 일반적으로 계획(PLAN)->개발(CODE)->빌드(BUILD)->테스트(TEST)->구현(RELEASE)->배포(DEPLOY)->운영(OPERATE)->모니터(MONITOR)의 순서를 가진다. 이 단계는 순환하며 모니터 이후 새로운 기능을 위한 계획부터 다시 시작하게 된다.

4. 애자일과의 차이점

애자일과 데브옵스의 가장 큰 차이점은 애자일은 고객과의 상호작용에 초점을 두는 반면 데브옵스는 개발팀과 운영팀간의 긴밀한 협력 관계 구축에 집중한다는 것이다. 또한 애자일은 반복과 고객의 피드백을 통한 지속적인 개선을 추구하지만 데브옵스는 지속적인 통합(CI)과 배포(CD) 그리고 통합과 배포를 위한 자동화(Automation)를 추구한다. 또한 통합, 배포 자동화를 위한 프로세스 및 도구간을 연결하는 파이프라인을 추구한다. 그리고 애자일은 핵심 가치를 바탕으로 여러 가지 해결 방법에 대한 갈래로 나뉘지는 반면 데브옵스는 핵심 목표를 향해 여러 가지 해결 방법들을 통합한다는 차이점이 있다.

III. 결 론

데브옵스(Devops)는 단순히 Dev(개발)과 Ops(운영)의 합성어는 아니라 생각한다. 또한 데브옵스는 데브옵스툴을 개발과 배포 전반에 적용하여 업무 효율을 향상 시키는 목적만에 한정된 것은 아니다. 크게 보면 하나의 가치를 위한 기업 문화의 변화라 볼 수 있다. 직원들의 협업과 개발, 운영, QA 조직의 최소화를 통한 이익 및 가치 실현에 목적이 있다. 이전까지의 개발 프로세스 방법론은 소프트웨어 개발 절차에 치중을 한 모델이라면 데브옵스는 개발부터 배포 및 서비스 운영 전반을 아우르는 모델이다.

참고 문헌

- 1) <https://www.redhat.com/ko/topics/devops>
- 2) <https://ko.wikipedia.org/wiki/%EB%8D%B0%EB%B8%8C%EC%98%B5%EC%8A%A4>
- 2) <https://engineering-skcc.github.io/devops/DevOps1>
- 4) <https://medium.com/dtevangelist/scrum-dfc6523a3604>
- 5) <https://medium.com/dtevangelist/scrum-dfc6523a3604>
- 6) <https://brunch.co.kr/@seonology/6>
- 7) http://fpost.co.kr/board/bbs/board.php?wr_id=122&bo_table=special
- 7) <https://www.programmersought.com/article/14752124332/>