

# DevOps의 성공적인 적용을 위한 가이드 라인 (2)

정두현 / ㈜씨에스리  
dhc97@naver.com  
컴퓨터시스템응용기술사  
정보시스템수석감리원

## DevOps적용을 위한 방법

### 1. DevOps를 통한 IT분야의 효과

- DevOps를 성공적으로 이루게 되면 실제로 IT투자 비용이 절감되고 생산되는 SW및 IT서비스가 실질적인 효과를 발휘하게 되어 잦은 SI비용의 절감과 낭비되는 요소를 줄일 수 있게 된다.
- 기본적인 IT 라이프 사이클은 SI 를 통해 신규개발하고 지속적인 SM에서 유지보수를 수행하고 진행하다가 효율이 떨어진다. 트렌드 반영을 위해서 SI를 수행하게 되면 다시 안정화하는데 SM이 노력을 쏟아야 하는 수고로움이 발생한다.
- 이런 라이프 사이클은 지속적으로 SM과 SI를 수행할 수 밖에 없는 환경을 만들게 되고 시스템 안정화의 노하우는 SI를 수행 할 때 마다 새롭게 바뀌는 트렌드를 찾기 위해 복잡하게 된다. 그래서 이러한 문제를 해결하기 위한 DevOps를 적용하려고 한다.
- 아래의 몇 가지 사례를 통해 DevOps의 성공 방향성을 찾아보려고 한다.

### 2. DevOps를 성공하기 위한 사례

#### 가. SM과 SI의 통합된 개발 조직 구성

- 보통 SM은 저렴한 비용으로 운영 하려 하고 SI는 단기적으로 고급 인력들 또는 개발 업체를 통해서 수행한다. 하지만 SI/SM을 관리적 입장에서 보게 되면 SI는 고급인력이 SM은 초/중급인력이 수행 하다 보니 방향성이 달라지고 SI를 통해서 만들려고 한 목적이 희석되는 경향이 있다.
- 몇몇 개발 조직들은 이러한 문제를 직시하고 개발조직을 통합하여 관리하는 사례가 있었다.
- 우선 SI프로젝트를 단기에 많은 인력투입에서 1년 단위로 평균인력 단위로 참여하도록 구성한다. 이렇게 평준화를 시키게 되면 1년간 계약이 가능하고 한번에 오픈 하는 것이 아닌 주기적으로 오픈을 할 수 있다. 즉 애자일의 작은 release의 개념과 유사하다.
- 개발 업체의 입장에서도 유리하다. 4~5개월 투입 후 끝나게 되면 인력은 이탈 하게 되고 다시 새로운 인력으로 SI를 수행하게 되면 연속성도 떨어지고 이전 개발의 노하우는 사라지고 신규개발과 같게 된다.
- 최근 SSPL(System & software Product Line)에서는 소스재사용을 넘어서 조직의 재사용을 언급 하고 있다. 조직의 중요성을 이해하기 시작한 것이다.

- 개발조직이 갖고 있는 특성을 그대로 반영하여 유사한 프로젝트나 기능을 구현하게 하는 것이다. 즉 유능한 개발자의 중요성을 파악하게 된 것이다.
- 즉 1년간의 계약하게 되고 1년마다 갱신하게 되면 유능한 인력을 유출 없이 개발이 가능하게 된다.(일부 기업들은 프로젝트가 끝나도 해당 개발자를 다른 업무에 투입하던가 관련 프로젝트(SM/SI)에 투입하여 이탈 없이 유지하려고 한다)

#### 나. 개발과 운영조직의 로테이션

- 모 SM개발 프로젝트는 약 100명이 넘는 외주 개발자로 구성되어 있다. SM이라고 하지만 실제로는 SI처럼 운영된다. 즉 각 개발자 별 혹은 팀을 만들어 해당 목표를 주어진다면 이를 구축하고 운영에 적용한다.
- 개발파트와 운영파트를 나누어 약 1달씩 운영업무를 맡게 된다. 즉 SI와 SM을 모두 수행하게 하여 서로간의 관점을 파악하게 되고 장애의 파급 효과 등을 경험하게 된다.
- 이런 조직의 특징은 일명 요구하는 것 만 해주는 개발에서 운영부분인 성능, 표준화 등을 파악하여 개발하게 된다.
- 이런 부분은 100명의 조직을 마치 한 팀처럼 구성하게 하여 개발과 운영의 문화적인 통합을 이루게 한다. 이런 조직은 자발적인 참여가 늘어나며 긍정적인 문화가 되었다.

#### 다. 모든 요구사항의 가시화

- 특히 DevOps팀은 가시화와 투명성이 매우 중요하다.
  - 즉 어떤 개발을 하고 있는지 그리고 각 개발간의 영향도가 있는지 파악이 가능하도록 Dashboard또는 칸반보드 형태로 메타를 적고 일정을 정리하여 자신의 진척도와 적용 시점을 파악하게 된다.
  - 이런 보드는 각 개발자 별로 자신의 업무의 상황을 파악하도록 하고 지시가 아닌 자발적인 개발이 가능하도록 유도할 수 있게 된다.
  - 궁극적인 목적은 개발 환경을 공유하고 서로 돕는 환경을 만드는 것이 중요하다.
  - 이전에 번다운 차트를 통해서 자신의 진척도를 가늠하고 스크럼 팀을 위해서 협업되고 자발적인 야근이나 철야가 진행되는 좋은 경험을 하게 되었다. 단, 이를 악용하는 경우도 있으니 유의가 필요하다
  - 악의적으로 이용할 경우는 자발적이 아닌 강제적인 분위기로 이끄는 경우 개발팀원들은 자기방어 형태로 완료를 늦추거나 빨리 개발이 가능한데도 천천히 개발할 수 있음으로 주의가 필요하다
  - 또한 피로도 문제가 있다. 즉 야근, 철야를 할 경우 반드시 휴식시간을 보장하는 것이 중요하다
  - 팀 리더나 PM수행을 해 본 사람이라면 누군가 주말 근무와 야근 또는 철야를 한경우 휴식을 보장하지 않을 경우 품질이 저하되고 전체적으로는 개발속도가 평균적으로 낮아지는 것을 경험 했을 것이다. (PM, 개발팀장 수행 시 후회한적이 많이 있었다)

#### 라. 적절한 도구의 활용

##### 1)이슈트래커

- 이슈 트래커나 버그 트래커 등은 이미 많이들 사용하고 있을 것이다. 그런데 문제는 그런 도구가 주체가 되는 것이 아니라 그런 도구를 통해 진행되는 절차와 각 구성원이 역할이 중요하다.
- 실제로 도구는 이러한 절차를 수행하기 위한 도구 일뿐 도구에 맹신하게 되면 일만 더 늘어나게 될 것이다.
- 이 슈트래커는 해야 할 일들을 이슈화 하거나 Task중심으로 열거하는 중요한 도구 이다.
- 이런 Task리스트는 실제 업무의 범위와 규모를 파악할 수 있게 된다.
- 단 이슈 트래커를 사용하기 위해서 파일럿 테스트나 실제 사용했던 조직의 경험을 바탕으로 처음부터 이슈트래커 중심으로 작업이 필요하다. 그렇지 않고 요구사항 정의를 별도로 한다면 이슈가 아닌 또 다른 개발 업무를 부여하는 것과 같게 됨으로 주의가 필요하다
- 특히 모든 채널은 단일 채널로 이용해야 할 것이며 하나의 도구가 선정되었다면 이를 다양하게 활용하여 오직 하나의 도구에 접속하면 모든 이슈와 개발 Task를 확인할 수 있게 되어야 한다.
- \* 필자는 보안환경이 아닌 경우는 Google Docs의 스프레드 시트를 이용하여 공유했었다.
  - 구글의 경우는 모두가 동시에 수정이 가능하며 이슈 공유도 가능함으로 즉시즉시 확인이 가능한 좋은 점이 있다
  - 단 중요 정보나 산출물 등은 외부에 노출될 수 있음으로 보안환경에서는 사용할 수 없다.

## 2)CI도구(Jenkins)

- 대부분 Jenkins를 사용하여 자동 배포를 하고 있다. 물론 Stage까지만 자동배포하고 운영계는 수동배포를 하는 경우도 많이 보아왔다. 운영계의 다운타임을 최소화 하기 위해서 수동배포를 하는 경우가 있으며 배포 시 발생하는 문제를 빠르게 확인할 수 있고 장애시간을 최소화 할 수 있는 장점이 있다.
- CI도구는 DevOps를 수행하기 위해서 당연히 활용되는 도구로 알고 있다.
- 하지만 이런 도구를 쓰기 위한 업무 프로세스부터 조율이 필요하다.
- 업무 프로세스를 만들기 위해서는 전체 흐름과 조직간의 역할을 볼 수 있는 관점이 필요하며 각 구성원간이 의견을 수렴하여 지속적으로 발전 시켜야 한다.
- CI는 SVN이나 GIT의 소스를 Repository에 저장한다. 그리고 이를 compile한다. 정상적으로 compile이 되었다면 해당 소스를 묶어서(Tar등) 해당 개발/QA서버에 올린다. 그러면 한번의 클릭으로 배포까지 자동으로 되게 된다.
- 또한 TDD를 활용하는 조직이라면 빌드 배포 시 자주 발생하는 오류를 검사할 수 있고 테스트 데이터까지 생성시켜 볼 수 있어서 매우 효율적이다. (단 TDD도 문화적으로 정착이 필요하다)
- CI도구를 활용하여 Daily Build및 Night Build를 수행하게 배포의 충격완화와 장애 위험이 줄어들게 된다. 하지만 운영에 적용하는 만큼은 모니터링이 필요함으로 적절한 운영 가이드가 필요하다. CI도구만 잘 활용해도 DevOps를 반쯤은 이루었다고 볼 수 있다.

### 3. 맺으며

위에서 2가지 도구와 조직구성의 방법론을 제시하였다. 첫번째 방법으로 문화만을 적용한다면 효율성은 있겠지만 직접적인 효과는 미비 할 것이고 두번째 방법으로 도구를 활용한다면 실질적인 업무는 완성하나 문화가 없다면 전통적 방법과 특별히 달라지지 않게 된다. 그럼으로 두 가지를 동시에 성공적으로 운영하는 것이 매우 중요하다.

다음시간에는 업무 프로세스를 중심으로 DevOps의 성공 방안을 제시하겠다.

궁금한 사항은 [dhc97@naver.com](mailto:dhc97@naver.com)으로 메일 주세요 “끝”

Contents connect communications!!

아이리포에 오시면 더 많은 지식을 가져가실 수 있습니다.

아이리포 온라인 : <http://www.ilifo.co.kr>

아이리포 지덤시리즈 : <http://www.jidum.com>

아이리포 IT지식창고 : <https://www.ilifo.co.kr/boards/knowledge>

아이리포 기술사/감리사 카페 : <http://cafe.naver.com/itlf>

서울시 마포구 상암동 1610번지, DDMC 3층 아이리포 교육센터

TEL: 02-303-9997 | MAIL: [ilifo@cslee.co.kr](mailto:ilifo@cslee.co.kr)