

## 신속 배포(Expedited Delivery)를 위한 데브옵스와 애자일 공존의 핵심고려 사항

2014. 6. 24. [제97호]

Contents

- ▶ 서론
- ▶ 데브옵스의 필요성
- ▶ 데브옵스 개념
- ▶ 애자일과 데브옵스
- ▶ 결론

## Key Message

지속적인 배포(Continuous Delivery)는 사용자에게 최대한 빠르고 효율적으로 새로운 기능을 제공하기 위한 소프트웨어 전략임. 이러한 지속적인 배포를 신속하게 처리하는 것을 신속배포(Expedited Delivery)라 하며, 데브옵스와 애자일은 신속한 제품생산 및 배포를 위한 상호 보완관계에 있음, 본 원고에서는 애자일 개발방법론에 내재한 데브옵스의 공통 개념에 초점을 두고, 제품의 신속배포를 위한 데브옵스와 애자일 공존 시 고려사항을 제시함.

## ▶ 서론

- IT조직에서 운영팀(Ops)은 시스템과 비즈니스부서 및 최종사용자간에 있어 직접적인 인터페이스 역할을 함
  - 따라서 운영팀(Ops)은 최종사용자들의 첫 번째 접점(Point of Contact)이라 할 수 있음
  - 운영 조직은 일반적으로 개발팀(Dev)으로부터 완성된 ‘배포가능한’ 산출물을 전달받아 제품으로 출시하는 릴리즈팀 뿐만 아니라 비즈니스 현장에서 유지보수를 제공하는 서비스팀을 포함함
- 운영팀은 효율적이고 제품에 대한 지식과 정보가 풍부해야지만 비즈니스 현장에서 만족할 만한 품질의 서비스를 제공할 수 있음
  - 효율적인 것은 완성된 실행가능한 산출물을 제품화하여 빠르게 배포하는 관점을 말하며, 제품관련 지식과 정보가 풍부하다는 것은 사용자의 질문을 이해하고 정확한 응답을 제공할 수 있는 것을 의미함
- 본 원고에서는 애자일 개발 프로젝트 상황 하에서 데브옵스 개념을 적용하여 데브옵스와 애자일이 신속한 배포(expedited delivery)를 위해 상호보완가능한 주요 고려사항을 살펴보고자 함

## ▶ 데브옵스의 필요성

- 종종 개발팀(Dev)이 솔루션을 만들 때, 기능적 요구사항에 온 정신을 기울이느라 배포 및 유지보수 측면의 요구사항을 이해하지 못하기도 함
  - 이로 인해 애플리케이션의 배포, 제품 유지보수, 재해 복구에 문제가 생길 수 있고, 이런 일이 반복된다면 기업은 IT조직을 책임감 없고 신뢰할 수 없는 조직이라고 여길 수 있음

- 애자일 개발 프로세스 상에서 개발팀은 모든 스프린트 막바지에 작동하는 기능 (working functionality)을 만들
  - 시스템 측면에서 완성된 기능은 릴리즈 일시가 도달하기 전까지 기다려야 하는데, 만일 릴리즈 날짜에도 운영팀이 통합, 배포 준비가 되지 않았거나 비즈니스 부서에서 새로운 기능을 사용할 준비가 되지 않았다면 릴리즈는 연기가 됨
  - 애자일의 주요 장점인 개발에서 시장으로의 출시까지의 최소한의 시간이 전혀 실현되지 못할 수 있음
- 데브옵스는 앞에서 기술한 내용에 대해 적극적으로 대응하는 많은 IT조직들이 도입하고 있는 모델임

## ▶ 데브옵스 개념

- 데브옵스는 비즈니스 관련부서, 개발부터(Dev), 운영부서(Ops) 등이 IT 솔루션을 목적인 시장 또는 현장에서 제시간에 이용가능하고 중단 없이 작동하도록 지속적으로 협업하는 상황을 일컫는 일종의 철학적 용어임
- 데브옵스는 자동화, 협업, 문화변동(cultural change)<sup>1)</sup>, 단순하며 효율적인 조직 구조 등을 요구하고 있음
- 또한, 조직, 사람, 프로세스, 툴 뿐 만 아니라 협업을 보장하는데 필요한 기술적 요소들과 요구 기능을 신속하게 제품화하는 데 있어 다른 이해관계자들과 협조하는 것을 강조함

## ▶ 애자일과 데브옵스

- 데브옵스는 애자일 방법론 하에서 개발된 제품을 신속하게 배포하여 빠른 주기의 개발이 가능한 애자일의 장점을 실현하게 함
- 다음 내용은 애자일 개발에 있어 데브옵스의 활동을 통해 신속한 배포 시 필요한 주요 고려사항들임

1) 문화변동(cultural change): 한 사회 또는 한 조직 내 문화가 내적으로 변동하거나, 다른 사회로부터 문화요소를 수용하고 변화하는 과정

### 1. 운영팀(Ops)의 지속적인 참여

- 데브옵스의 기본적인 요구사항은 운영팀이 솔루션 개발 생애주기에 걸쳐 개발팀과 지속적으로 교류하는 것임
- 운영팀은 초기 개념화 단계(vision stage)부터 참여해서 비즈니스 목적, 개발 과정, 릴리스 단계까지 이해해야 함
- 운영팀은 솔루션의 기술적, 일정상의 실현가능성을 결정하는데 기여해야함
- 개념화 단계에서 개발 단계에 걸쳐, 운영팀은 개발팀이 운영관련 요구사항을 만들고 검증하는데 필요한 투입요소(input)들을 제공해야 함

### 2. 제품 소유자(PO: Product Owner)의 역할 확립

- PO는 개발팀(Dev)에 있어 비즈니스 목적 달성을 위해 사용자 입장에서 우선순위를 결정하고 비즈니스 프로세스를 검토하는 제품 책임자임
- PO는 제품에 대한 비전을 가지고 있고 충족해야할 기능적 요구사항에 대한 완벽한 통찰력을 가져야 함
  - 그러나 PO가 비기능적 요구사항(NFRs: nonfunctional requirements)를 잘 이해하지 못하거나 잘 설명하지 못한다면, 개발팀은 아키텍처를 만들거나 최종 솔루션을 만들면서 이 요구사항을 제대로 반영하지 못할 것임
- IT팀과 비즈니스 부서 리더들은 다음과 같은 기술관련 요구사항에 따라 PO를 NFRs에 기본적인 공감을 갖도록 준비시켜야 함
  - 배포와 유지보수 플랫폼
  - 인프라스트럭처를 유지하는 벤더 파트너들에 대한 의존성
  - 최종 솔루션에 필요한 인터페이스와 써드파티(Third-party) 애플리케이션
- PO가 모든 부문의 전문가가 될 수는 없으나, 기능적 요구사항과 비기능적 요구사항을 모두 예상할 수 있고, 이것들을 IT와 비즈니스 사이에서 적합한 소통자가 되어야 함

### 3. 제품 백로그

- 일반적으로 제품 백로그는 기능 요구사항과 관련하여 개발과정이나 개발과정의 흐름에 집중함
- PO와 개발팀은 브레인스토밍, 개발과정의 분할, 기능 요구사항의 문서화에 잘 훈련되어 있음
- 그러나 종종 NFRs은 백로그에 제대로 상세화되지 않기도 함에 따라, 기능 요구사항

과 더불어 백로그는 다음과 같은 사항들을 기술해야 함

- 성능 요구사항(Performance requirement)
- 배포와 유지보수와 관련된 기술적 요구사항
- 빠른 롤백(Rollback)과 롤포워드(Roll forward)의 가이드라인을 개발하기 위한 요구사항
- 보안/방화벽 요구사항

#### 4. 개발팀내의 운영팀 대표자 확보

- 애자일 개발팀은 다기능(Cross-functional)적이고 자가조직(self-organizing)적임
- 이러한 팀도 운영팀 인력을 포함해야 하는가에 대한 질문에 대해서는 교차기술(Cross-skilled)<sup>2)</sup>을 지닌 팀원에게 얼마나 의존하는냐에 달려있음
- 일반적으로 스타트업 IT 조직에서는 개발자와 테스터의 일부는 배포와 레벨3(버그 수정) 유지보수에도 책임이 있음
  - 배포와 유지보수의 요구사항은 계획과 리뷰 회의에서 제대로 토론되어야 함
- 큰 조직의 운영팀은 여러 개발 팀들로부터 배포받은 완성된 코드를 인수하는 전담 팀이 필요한데, 이러한 경우 개발팀과의 업무 교대는 좋은 대안이 될 수 있음
- 즉, 일부 개발자들이 특정기간동안 운영(Ops) 역할을 수행하고, 적성이 맞는 일부 운영팀원은 특정기간동안 개발팀 중 일부(Dev)가 될 수 있음
  - 이 방식은 개발 주기 동안에 개발팀 내에서 운영팀을 제대로 대표할 수 있음

#### 5. 완료의 정의

- 개발주기에 운영을 포함하는 또 다른 주요 수단은 완료의 정의에 운영관련 관점을 엮는 것임
- 표준 코딩, 테스트, 요소 문서화에 따라 배포 플랫폼에서 코드 검증, 문서화의 일부 로써 상세 유지보수 지시서, 이 지시들의 요약서들을 완료의 정의에 포함해야 함
- 여기에도 또한 운영팀의 투입이 중요함

2) 교차기술(Cross-skilled) : 이종의 서로 다른 다양한 업무를 처리하는 작업 역량, 본 보고서에서는 '개발기술과 운영기술을 모두 소유한'것으로 이해

## 6. 스프린트 계획과 일일 스탠드업(Daily Stand-up)

- 스프린트 백로그 계획과 일일 스탠드업에서 백로그 항목을 우선순위화하고 프로세스를 논의하는 동안에 운영 니즈에 주목해야 할 필요가 있음
- 운영 관점에서 토론하는 스프린트 계획과 선택된 일일 스탠드업에 운영 팀을 포함시키는 것은 좋은 아이디어임
- 인프라 공급자와 시스템 통합자에 대한 의존도는 운영팀을 투입하는 이 단계에서부터 고려되어야 함

## 7. 스프린트 리뷰

- 운영팀이 개발 주기에 지속적으로 참여할 때, 운영팀 또한 스프린트 리뷰의 일부가 되어야 한다는 것을 말할 필요도 없음
- 개발팀은 솔루션의 운영관련 특성들을 제시해야 하는데, 모든 스프린트 리뷰에 운영관련 특성들을 포함하지 않더라도, 운영 측면에서 무슨 일이 발생할지를 파악하고, 제품을 향상시키고 운영 요구사항을 확실히 포함하는 등 다음 스프린트에서 이러한 투입요소를 제공해야함
- 만약 운영팀이 독립적이라면 개발팀은 제품 보증에 운영팀을 확실히 참여시켜야 함

## 8. 스크럼 오브 스크럼(Scrum of Scrums)

- 여러 스크럼 팀들이 솔루션을 작업할 때는 각 팀의 산출물 통합을 주의 깊게 계획하고 조정하여 실행하는 스크럼 오브 스크럼이 요구됨
- 각 스크럼 팀은 운영 요구사항을 고려해서 요구사항에 맞는 특성들을 만들어야 함
- 제품소유자는 최종 제품, 여러 스크럼 팀들이 어떻게 개발을 진행할 것인지와 어디서 어떻게 배포할 것인지에 대한 견해를 가지고 있어야 함
  - 각 스크럼 팀을 위해 특정 투입요소를 제공하는 운영팀을 참여시켜야 함
  - 스크럼 오브 스크럼에 PO가 참여해서 각 스크럼 팀들이 수립한 계획과 실행에 운영팀을 포함했는지를 확인해야 함

## 9. 계획 정렬(alignment)

- 데브옵스와 애자일은 매우 상호보완적이고 비즈니스 부서와 릴리즈 팀이 연간 릴리즈 일정을 계획하는 것에 기여함

- 개발팀과 지속적인 교류와 협업을 통해, 운영팀은 어떤 기능들이 언제 완성되는지 개발일정을 알아야 함
- 이러한 통찰력과 스프린트 완성 패턴을 활용하여, 운영팀과 개발팀은 잠재적인 릴리즈 일정을 합리적으로 조정해야함
- 이러한 일정 조정이 일어날 때, 유지보수 팀은 완성된 기능을 짧은 시간에 빠르게 제품화하여 출시 할 수 있으며, 이를 통해 애자일의 주요 장점이 실현됨

## 10. 지표(Metrics)

- 데브옵스가 신속한 릴리즈에 얼마나 도움이 되는지를 측정하기 위해서는 다음과 같은 몇 가지 지표들을 사용할 수 있음
- 다음의 지표는 데브옵스와 애자일을 사용하는 조직의 배포상의 효율성을 증명하고 관리할 수 있도록 함
  - 릴리즈 날짜 엄수 비율
  - 릴리즈 수의 증가 비율
  - 제품까지 릴리즈 되는데 걸리는 시간
  - 비기능적 요구사항(NFRs)의 충족 비율

## ▶ 결론

- 본 원고에서는 데브옵스를 이행하는 매우 특정한 애자일 스프린트 관점에 집중하였음
- 그러나, 데브옵스는 훨씬 거대하고 비즈니스, IT, 벤더 관계자들을 모두 아우르는 다양한 요소를 포함함

## 참고 자료

1. <http://scaledagileframework.com/devops/>
2. <http://www.skytap.com/blog/combining-devops-with-agile-development>
3. <http://www.slideshare.net/xebialabs/optimizedevopsandagilestrategieswithdeploymentautomation>
4. <http://www.agileweboperations.com/lean-agile-devops-related>