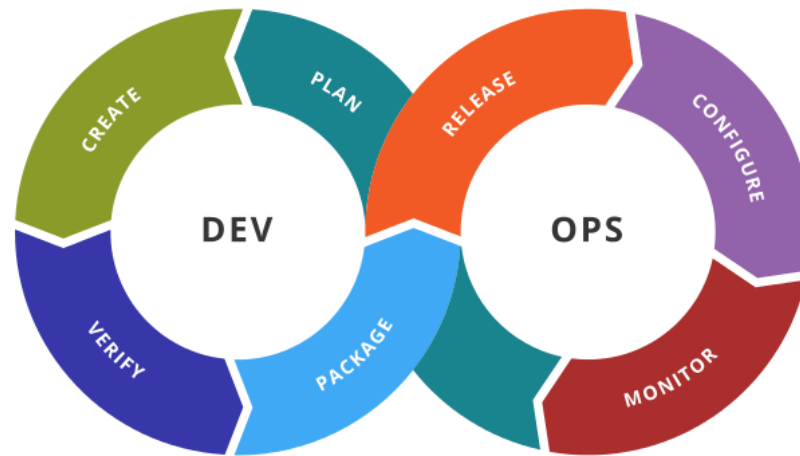


Time goes now



DevOps: The way to go



resisa@gmail.com

2018.02



If I sleep now I will have a dream, but if I study now I will make my dream come true ...

❖ 데브옵스의 실체 ???

"DevOps is development and operations **collaboration**"

"DevOps is treating your **infrastructure as code**"

"DevOps is using **automation**"

"DevOps is feature **switches**"

"DevOps is **small** deployments"

"**Kanban** for Ops?"

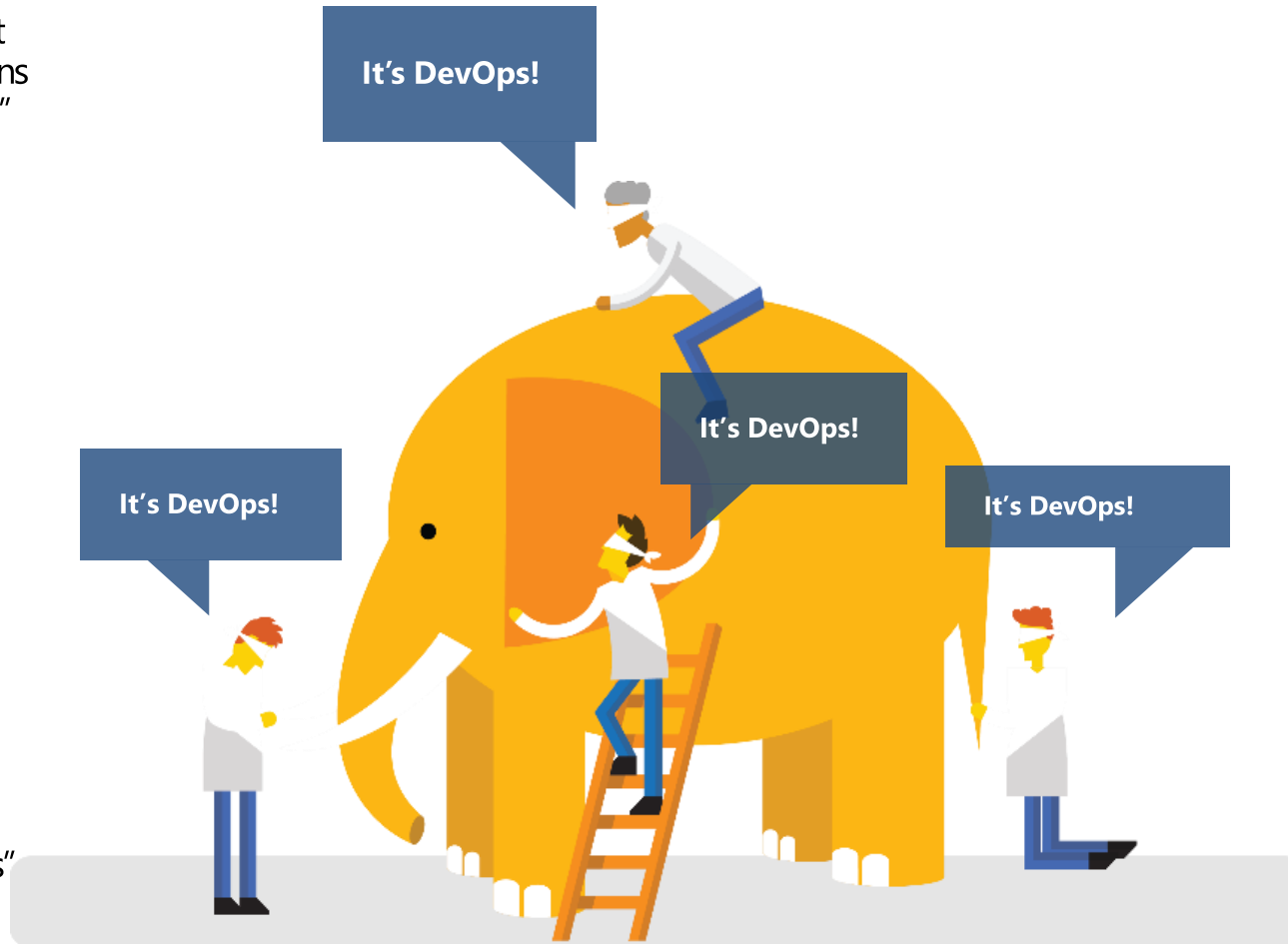


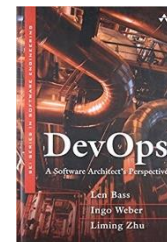
Image From "The Microsoft DevOps Story"

❖ Definition #1 (from Wikipedia.org)

데브옵스(DevOps)는 소프트웨어 개발(Development)과 운영(Operations)의 합성어로서, 소프트웨어 개발자와 정보기술 전문가 간의 소통, 협업 및 통합을 강조하는 **개발 환경이나 문화**를 말한다.

데브옵스는 소프트웨어 개발 조직과 운영 조직간의 상호 의존적 대응이며, 조직이 소프트웨어 제품과 서비스를 빠른 시간에 개발 및 배포하는 것을 목적으로 한다.

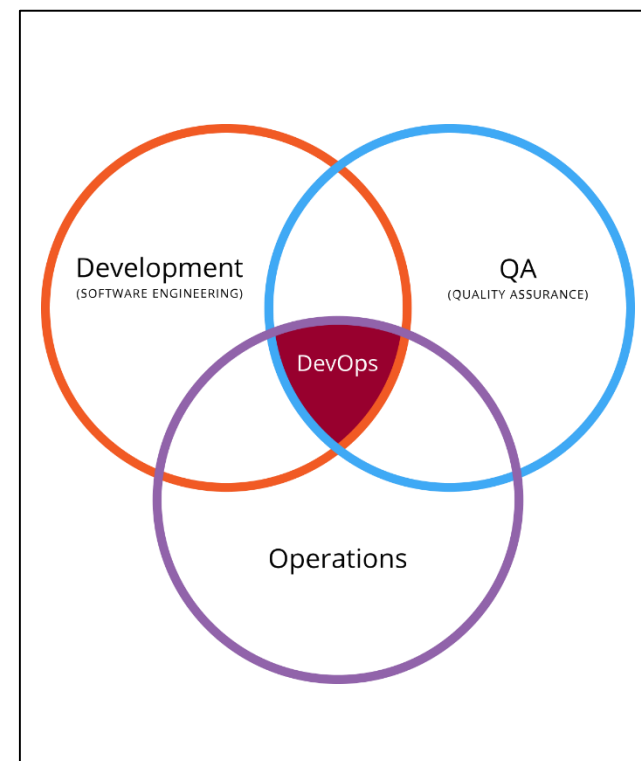
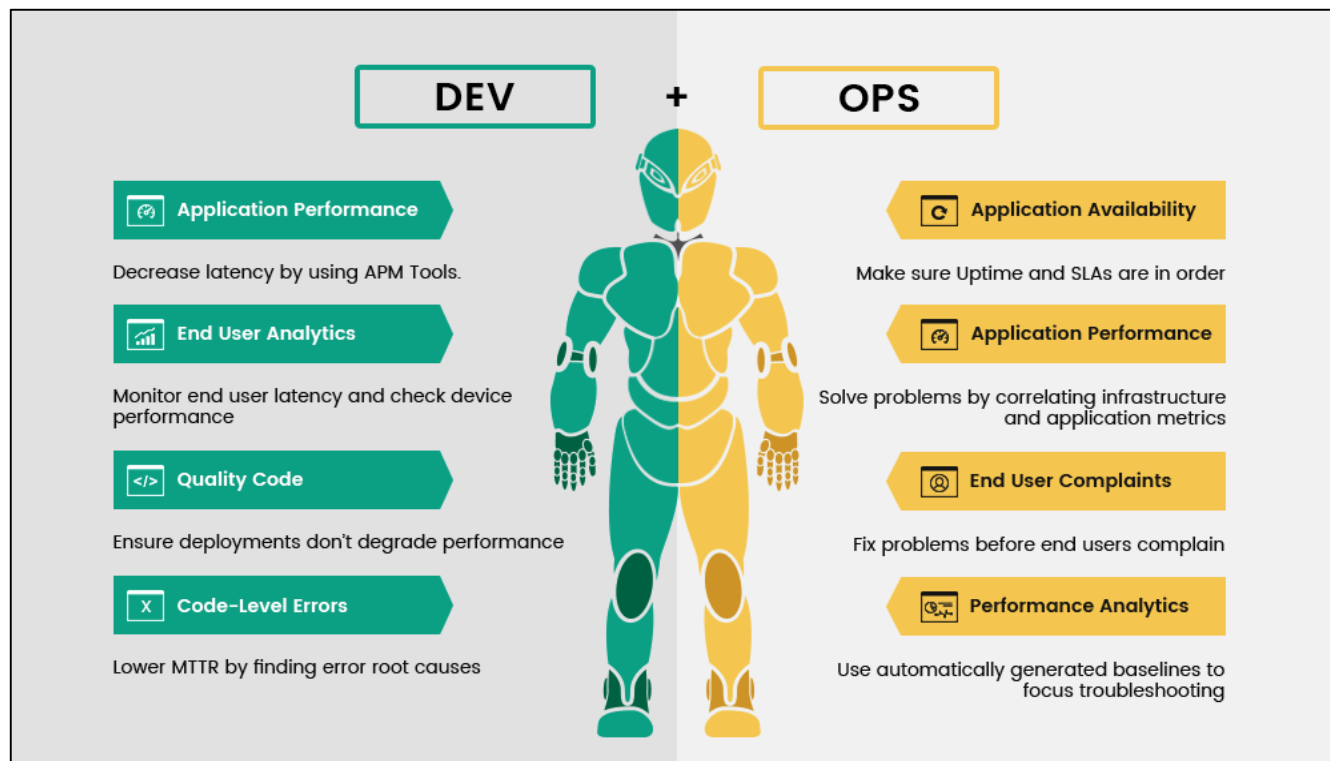
❖ Definition #2 (From DevOps: A Software Architect's Perspective)



데브옵스(DevOps)는 높은 품질을 유지하면서, 시스템에 대한 변경 사항의 적용 및 그 변경 사항을 일반적인 생산 환경에 적용하는 시간을 줄이기 위한 일련의 **실천방법(Practices)**이다.

❖ 데브옵스란 ...

- 개발(**Dev**elopment) + 운영(**Ops**)
- 프레임워크나 방법론이 아니다
 - ✓ 다양한 프레임워크와 방법론을 적용하고 활용한다.
- 개발 철학 + 구현 환경(플랫폼, 툴셋 ...) = 개발 문화



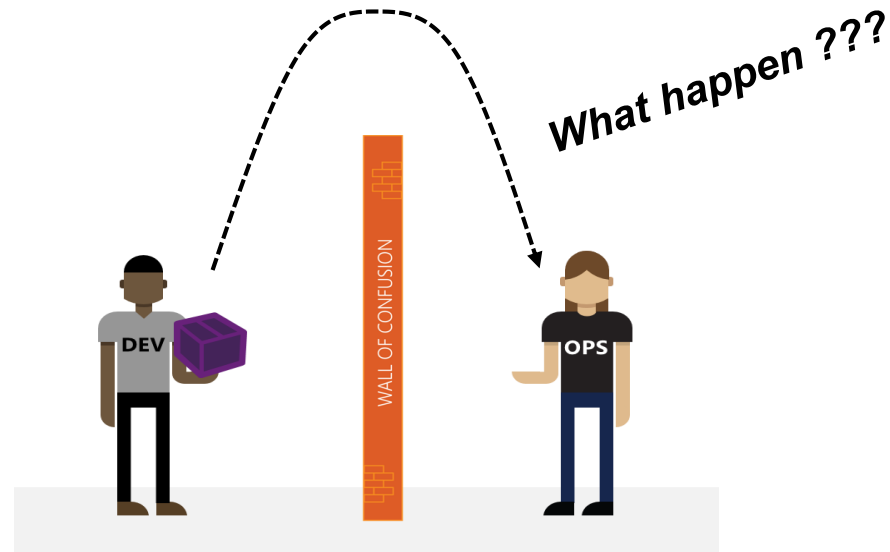
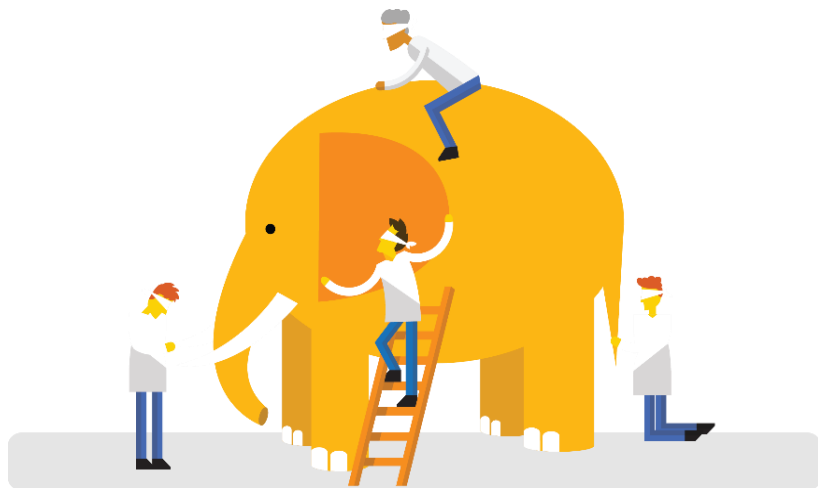


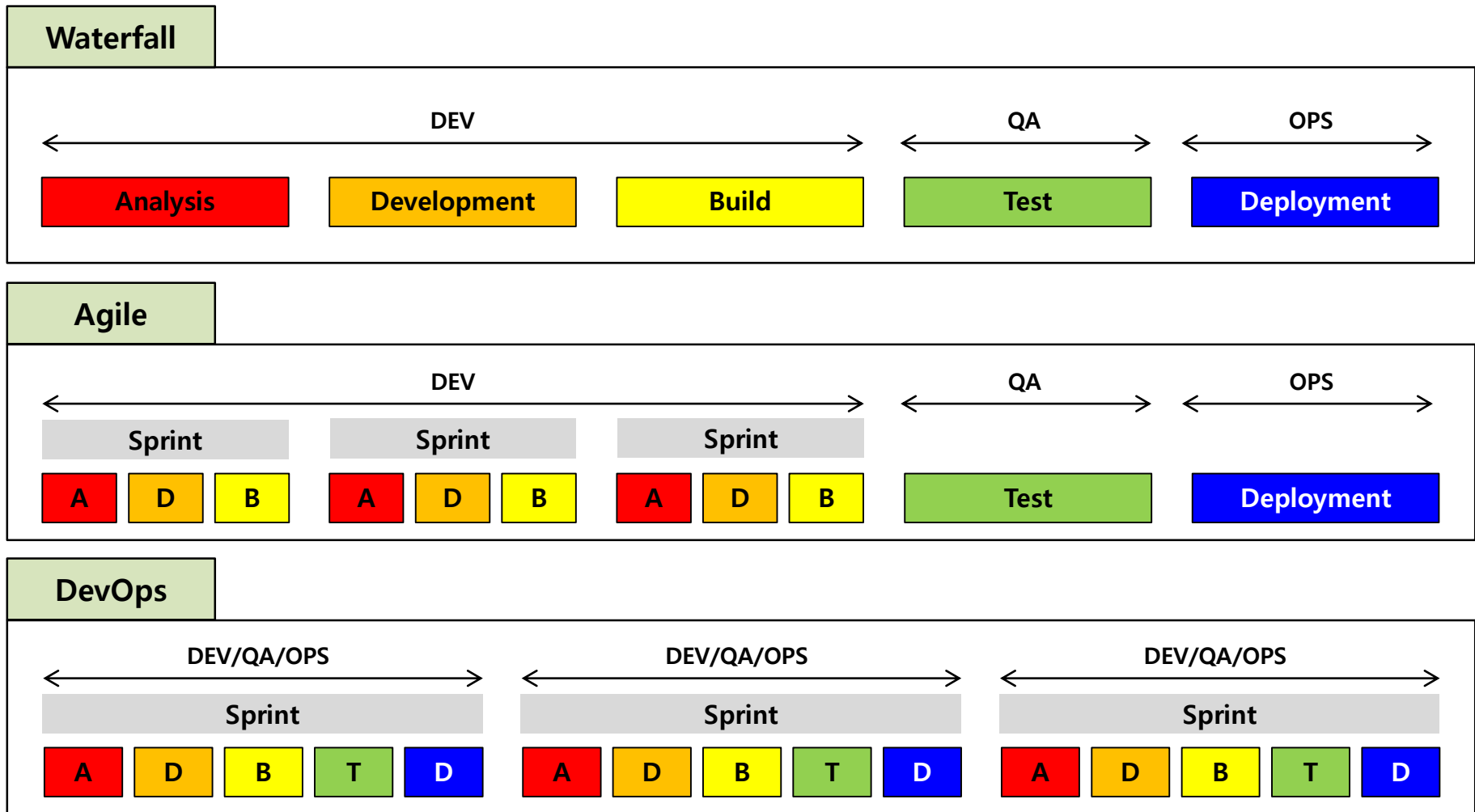
Image From "The Microsoft DevOps Story & Atos Consulting CH"

❖ 데브옵스 구현 시 필요사항

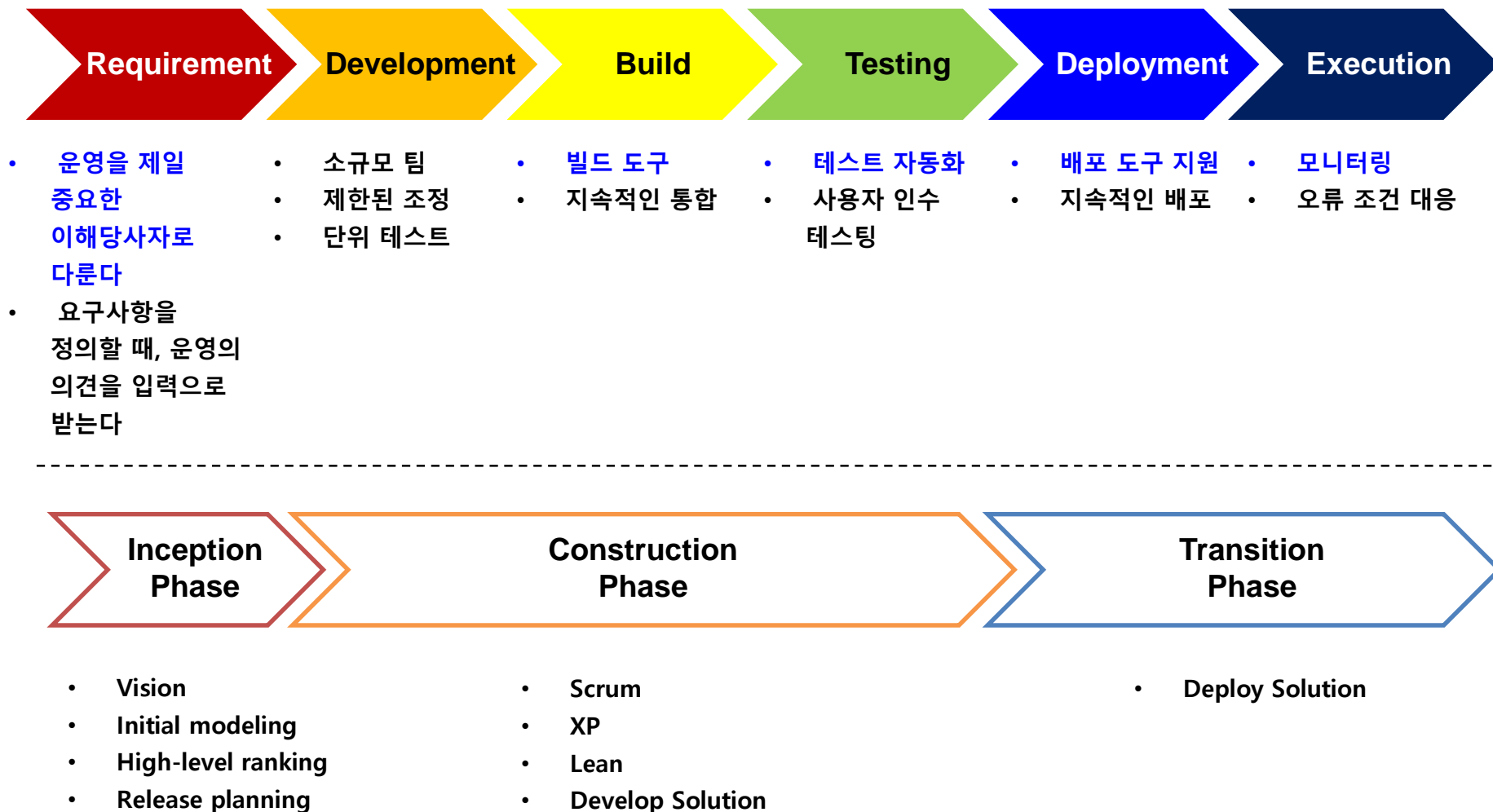
- 애자일 및 린 소프트웨어 개발의 실천방법
- 애자일 및 린 서비스 관리 실천방법
- 가상화 및 클라우드 인프라스트럭처(내부/외부)
- 데이터 센터 자동화 및 구성 관리 도구
- IAC (Infrastructure as Code)
- 모니터링 및 자기 복구 기술



❖ 기존 방법론과의 차이

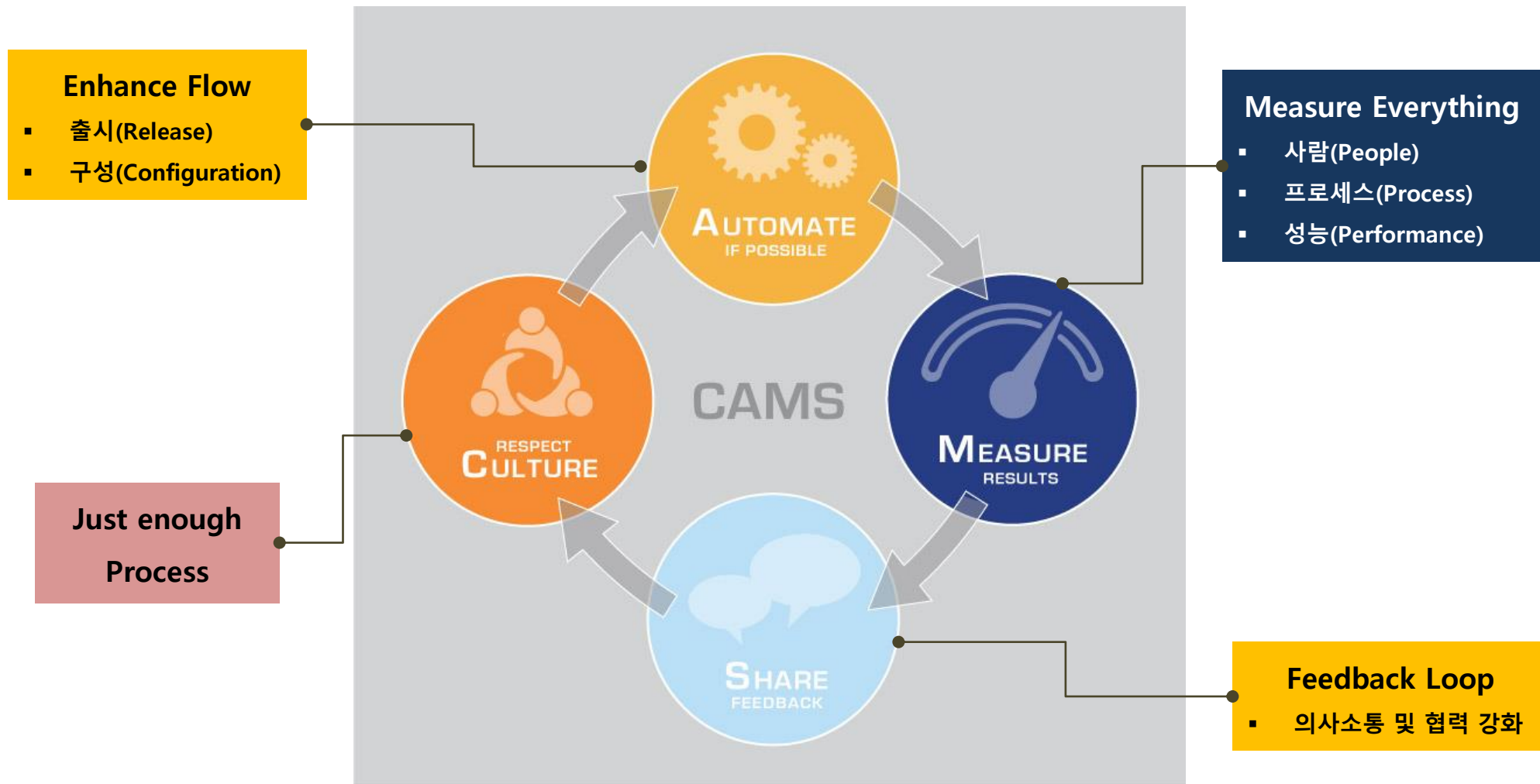


❖ 데브옵스 수명주기



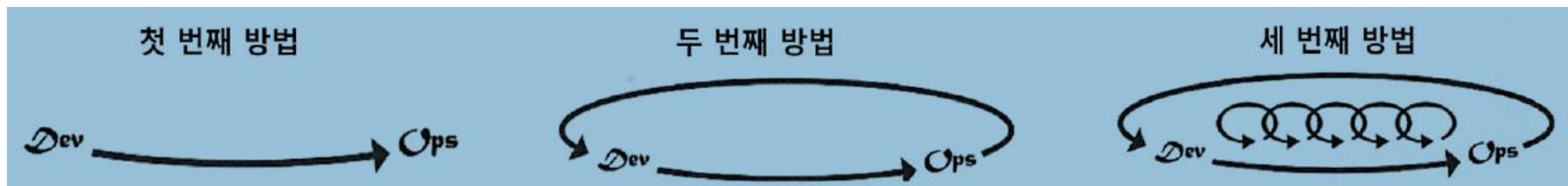
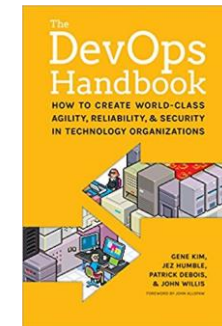
❖ 데브옵스의 가치

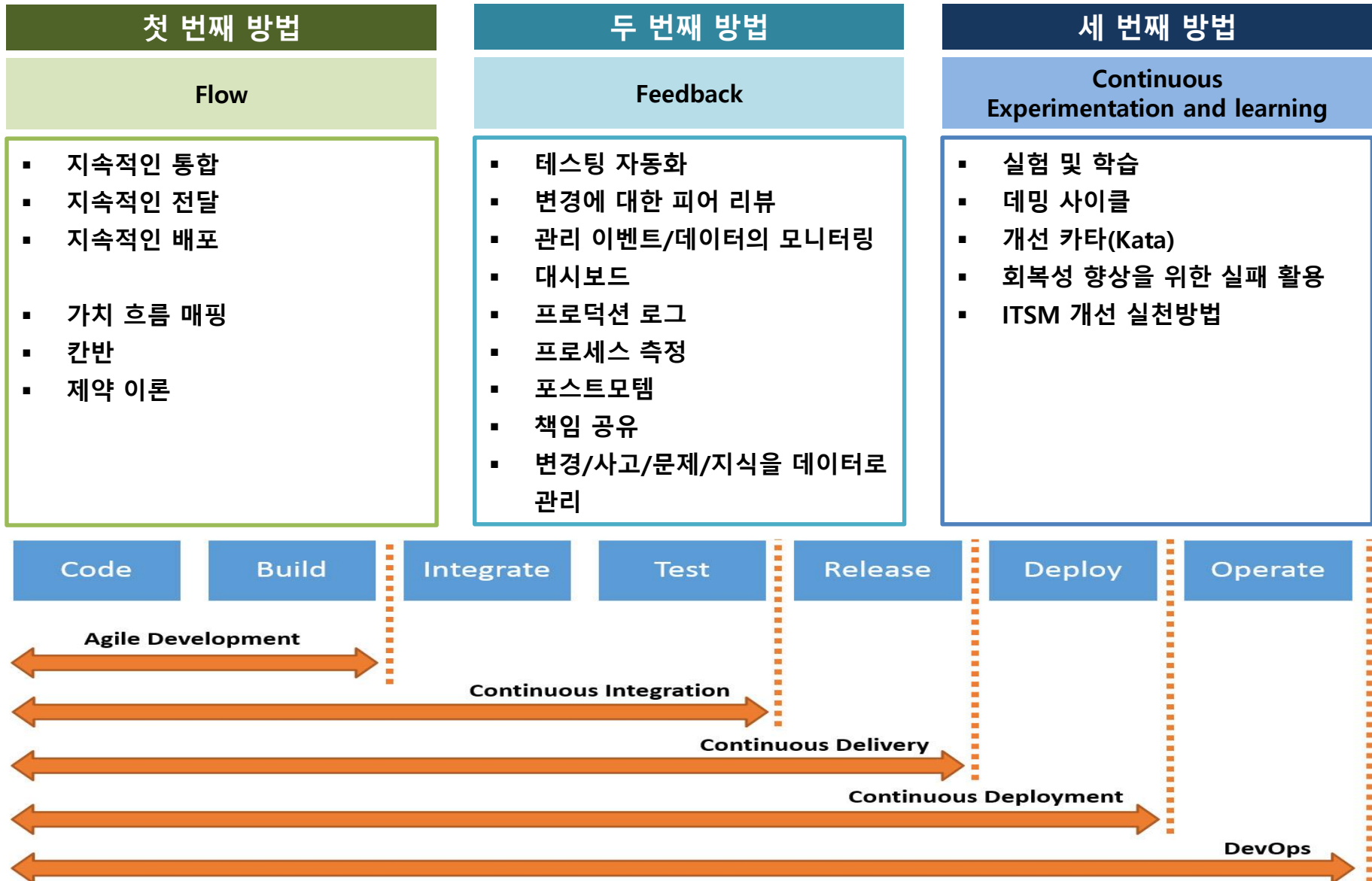
- CAMS (Culture, Automation, Measurement and Sharing)



❖ 데브옵스 원칙 - 세 가지 방법

- "The Phoenix Project"에서 소개
- 첫 번째 방법 - 흐름(Flow)
 - ✓ 작업의 흐름을 이해하고 증가시켜라 (Left to Right)
- 두 번째 방법 - 피드백(Feedback)
 - ✓ 피드백 루프 생성
 - ✓ 지속적인 통합 활성화 (Right to Left)
- 세 번째 방법 - 지속적인 실험 및 학습
 - ✓ 공유 문화 생성
 - 실험, 위험 감수, 실패로부터의 학습
 - 반복과 실습이 숙달의 전제 조건임을 이해





❖ 팀 규모 - 피자 두 판의 법칙

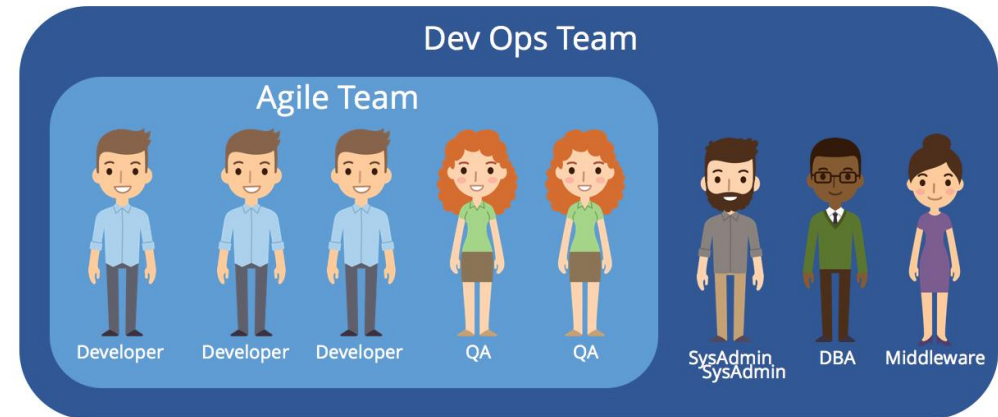
■ 소규모 팀의 장점

- ✓ 빠른 의사 결정
- ✓ 응집성
- ✓ 의사소통



■ 역할(Roles)

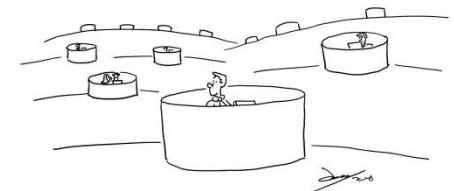
- ✓ 팀 리더와 팀 멤버
- ✓ 서비스 담당자(Service Owner)
- ✓ 신뢰성 엔지니어(Reliability Engineer): 모니터링, 문제 분석 및 진단
- ✓ 게이트키퍼(Gatekeeper): 배포 파이프라인에서 다음 단계로의 이동을 결정
- ✓ 데브옵스 엔지니어(DevOps Engineer): 구성 관리





❖ 데브옵스 적용의 장애물

- 조직의 문화와 형태
 - ✓ 특정 도메인(ex: 규제 분야)에 속한 조직은 문제 발생 위험 감소가 우선
 - ✓ 특정 조직은 오류보다 민첩성과 빠른 대응이 우선
- 부서 유형
 - ✓ 부서마다 인센티브가 다름(개발 - 빠른 개발, 운영 - 안정 유지) → 문화 충돌
- 사일로 심리(Silo Mentality)
 - ✓ 공통의 목표가 항상 모든 팀을 하나로 만들지 않는다 → 자신의 팀 > 다른 팀
- 도구 지원
 - ✓ 자동화 도구의 설치/구성/사용에는 전문성이 필요하다
 - ✓ 모든 개발팀이 공통 프로세스를 따르는지 확인할 수 있어야 한다
- 개인적인 문제
 - ✓ 업무의 가성비 문제: 운영 업무를 개발이 한다면, 좋은 결과만 있는가?



❖ 데브옵스 적용 효과

- 더 빠른 적시 출시
- 새로운 출시 시, 더 낮은 실패 비율
- 짧아진 리드 타임
- 더 빨라진 평균 배포 시간

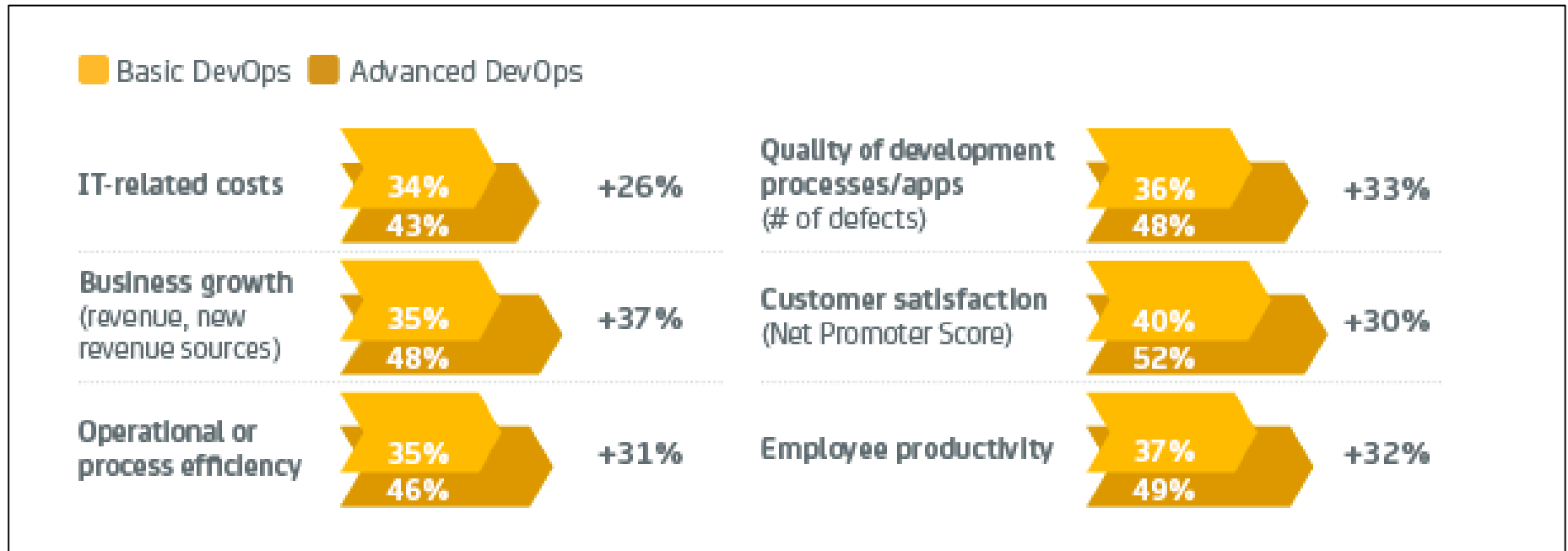


Image From "Cygnit Infotech & CA Technologies"

❖ 데브옵스 적용 가이드

■ 데브옵스를 시작하려면 ...

- ✓ 사람들이 올바른 일을 쉽게 할 수 있도록 하라
- ✓ 단순함 - 나쁜 프로세스를 자동화 하지 마라
- ✓ 높은 가치를 갖고 반복되는 작업을 자동화하라
- ✓ 오류를 증명할 수 있는 작업을 자동화하라
- ✓ 워크플로우 병목지점과 의사소통 흐름을 최적화하기 위해 자동화하라
- ✓ 자동화 모니터링과 통보 실천방법을 향상시켜라



■ 데브옵스 문화를 적용하려면 ...

- ✓ 올바른 사람들이 함께 모이도록 하라
- ✓ 모든 사람이 같은 사항을 알 수 있도록 하라
- ✓ 지속적인 변화를 유도하는 역량을 구축하라
- ✓ 중요한 행위에 초점을 맞춰라
- ✓ 실험하고 학습하라
- ✓ 이익을 통합하고, 더 많은 변화를 발생시켜라
- ✓ 관성을 방지하라.



