

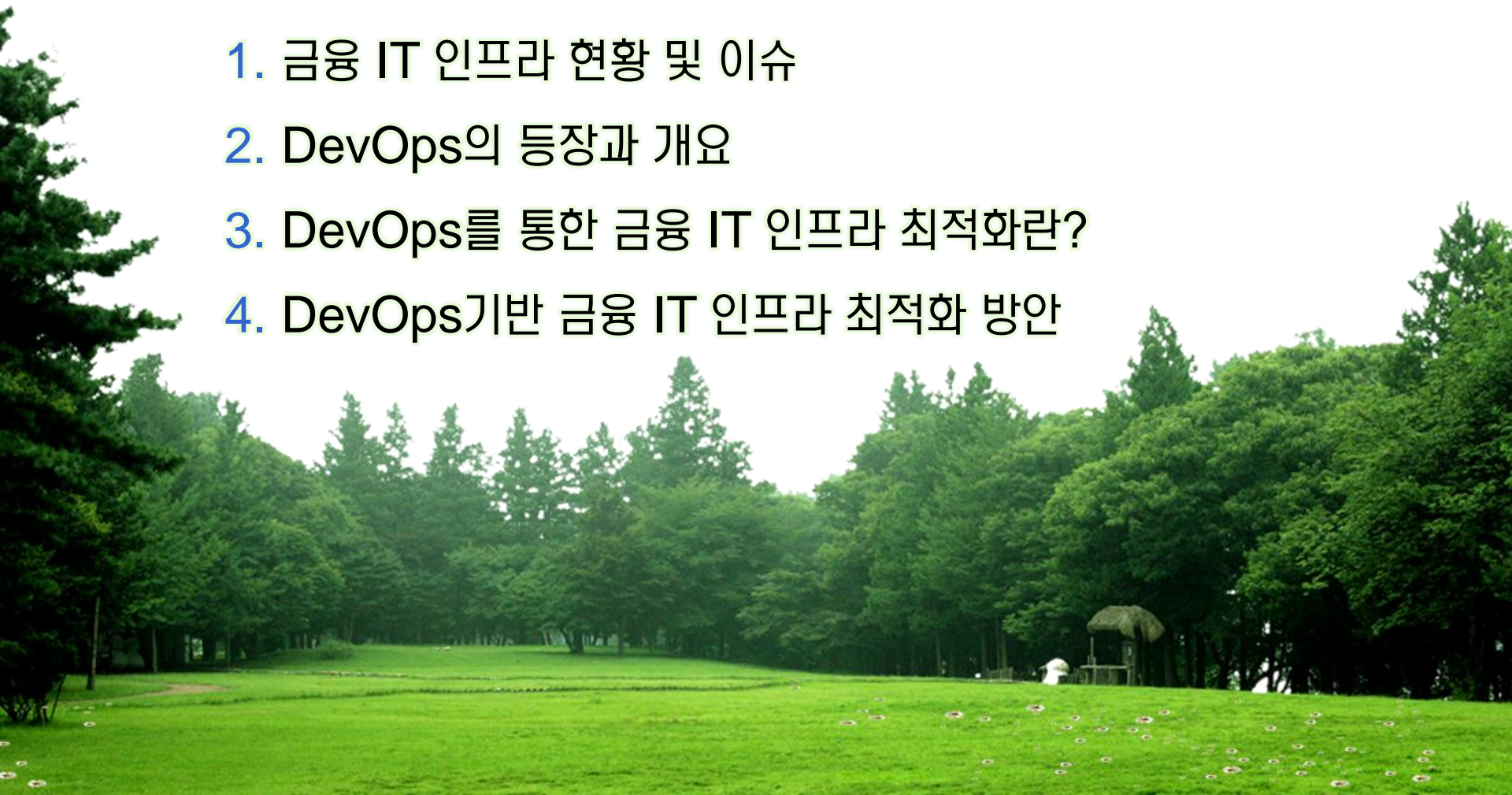
2014년 전망, 금융 IT Innovation 컨퍼런스

DevOps기반의 금융 IT인프라 최적화 방안

이현찬 (hcleee@kr.ibm.com), IBM Rational
IT DevOps/ALM/Agile Specialist

AGENDA

1. 금융 IT 인프라 현황 및 이슈
2. DevOps의 등장과 개요
3. DevOps를 통한 금융 IT 인프라 최적화란?
4. DevOps기반 금융 IT 인프라 최적화 방안



Agenda

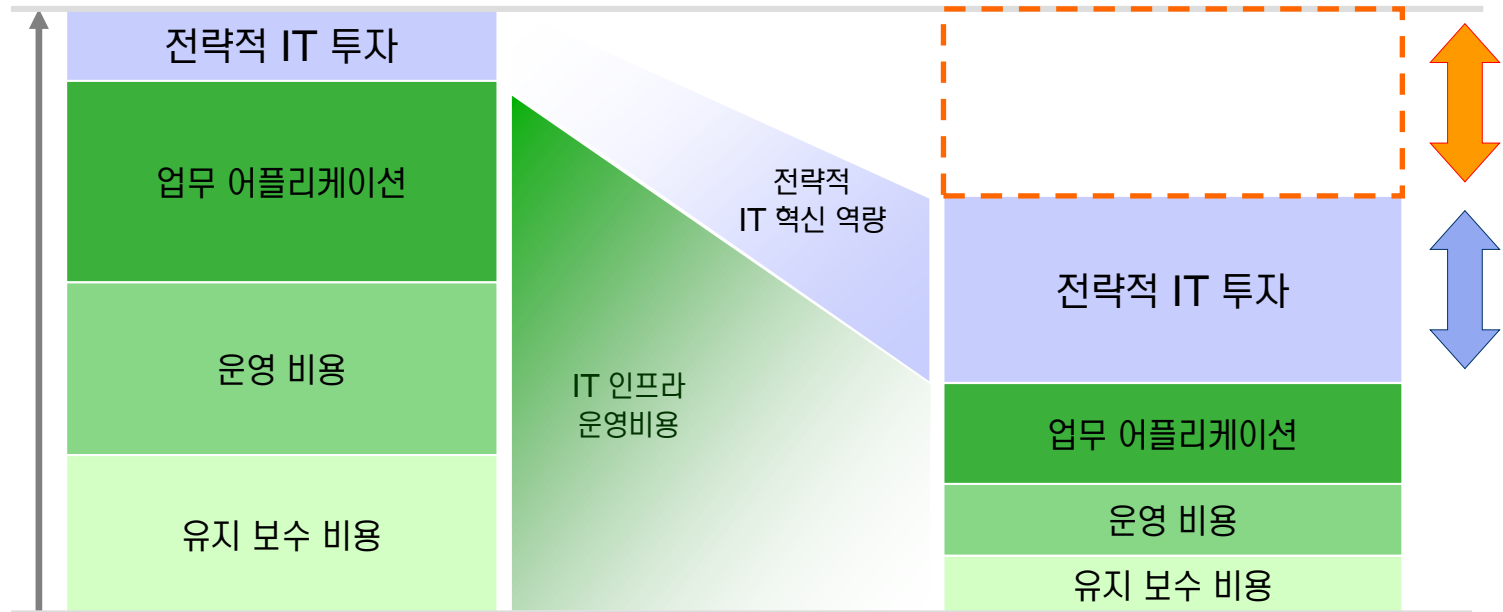
1. 금융 IT 인프라 현황 및 이슈

Contents

- 새로운 패러다임 등장
- 복잡해져만 가는 IT 인프라
- 현 IT 구성의 한계와 이슈

새로운 패러다임 – Do More with Less

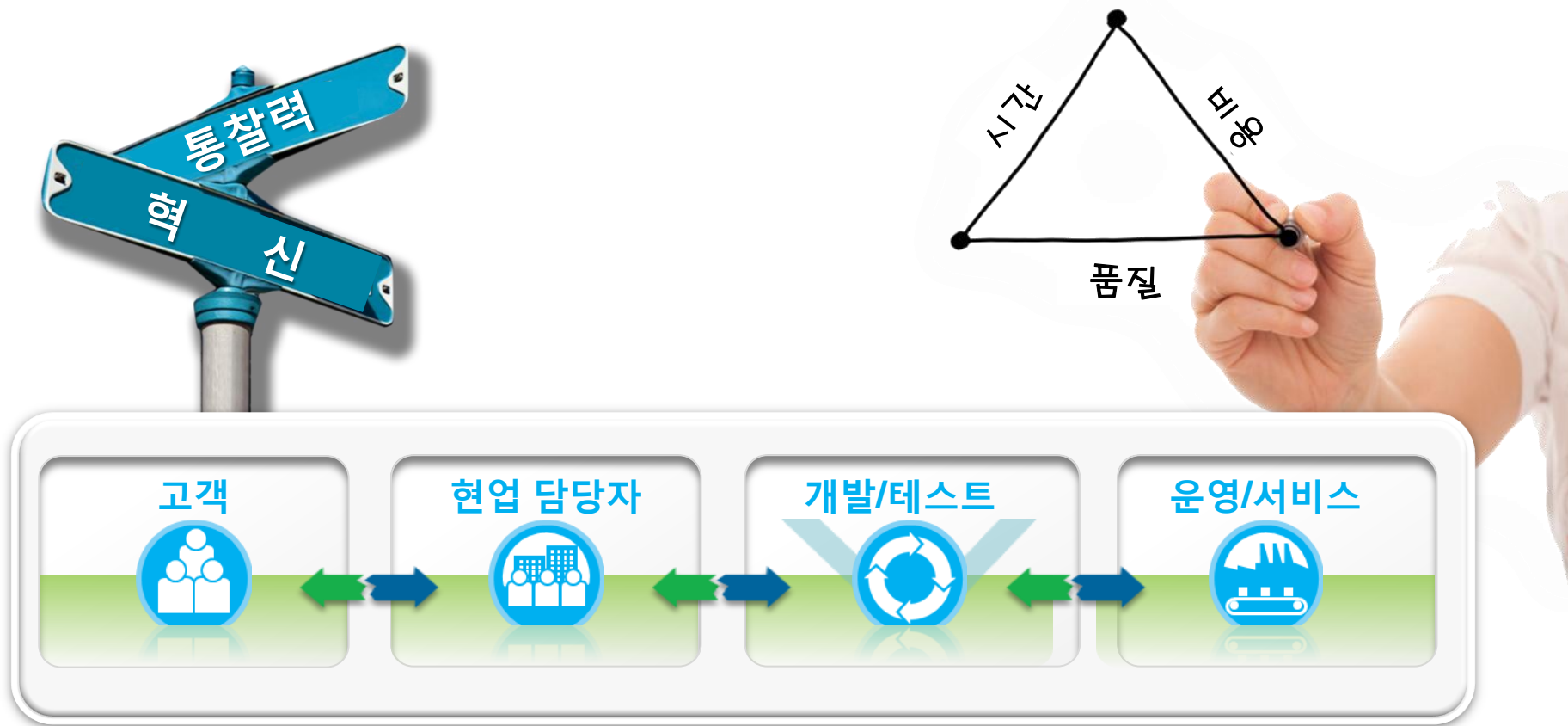
[중장기 측면에서 본 혁신적 IT 개선전략]



Source: Forrester, "Application Modernization and Migration Trends in 2009/2010"

주어진 예산과 리소스를 통해 빠르게 변화하는 비즈니스에 대응하기 위해서는
작은 것으로 더 많은 것을 할 수 있어야 합니다.

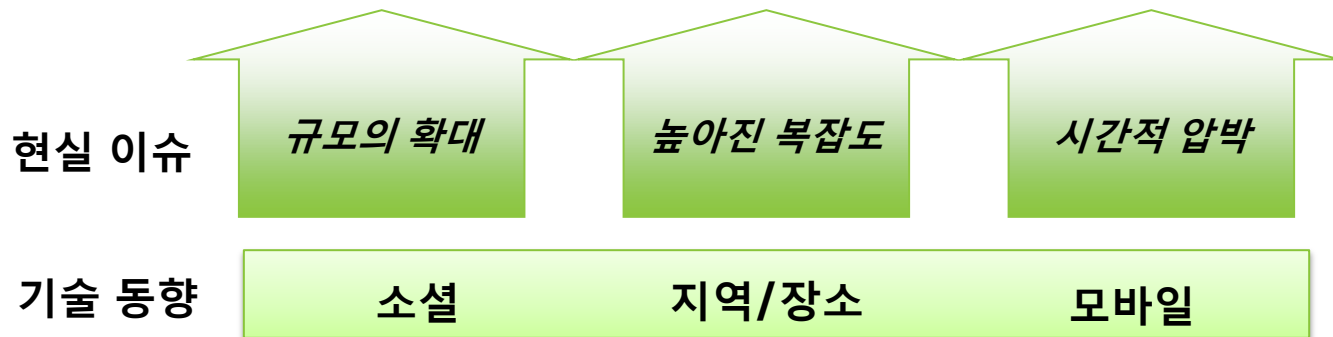
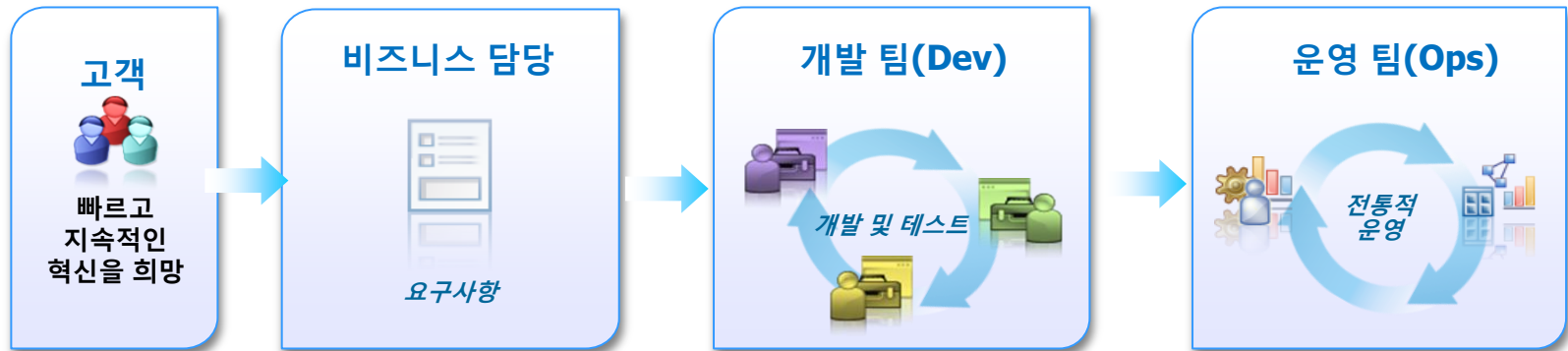
시장 과제 및 고객 피드백에 대한 신속한 대응이 성패를 결정



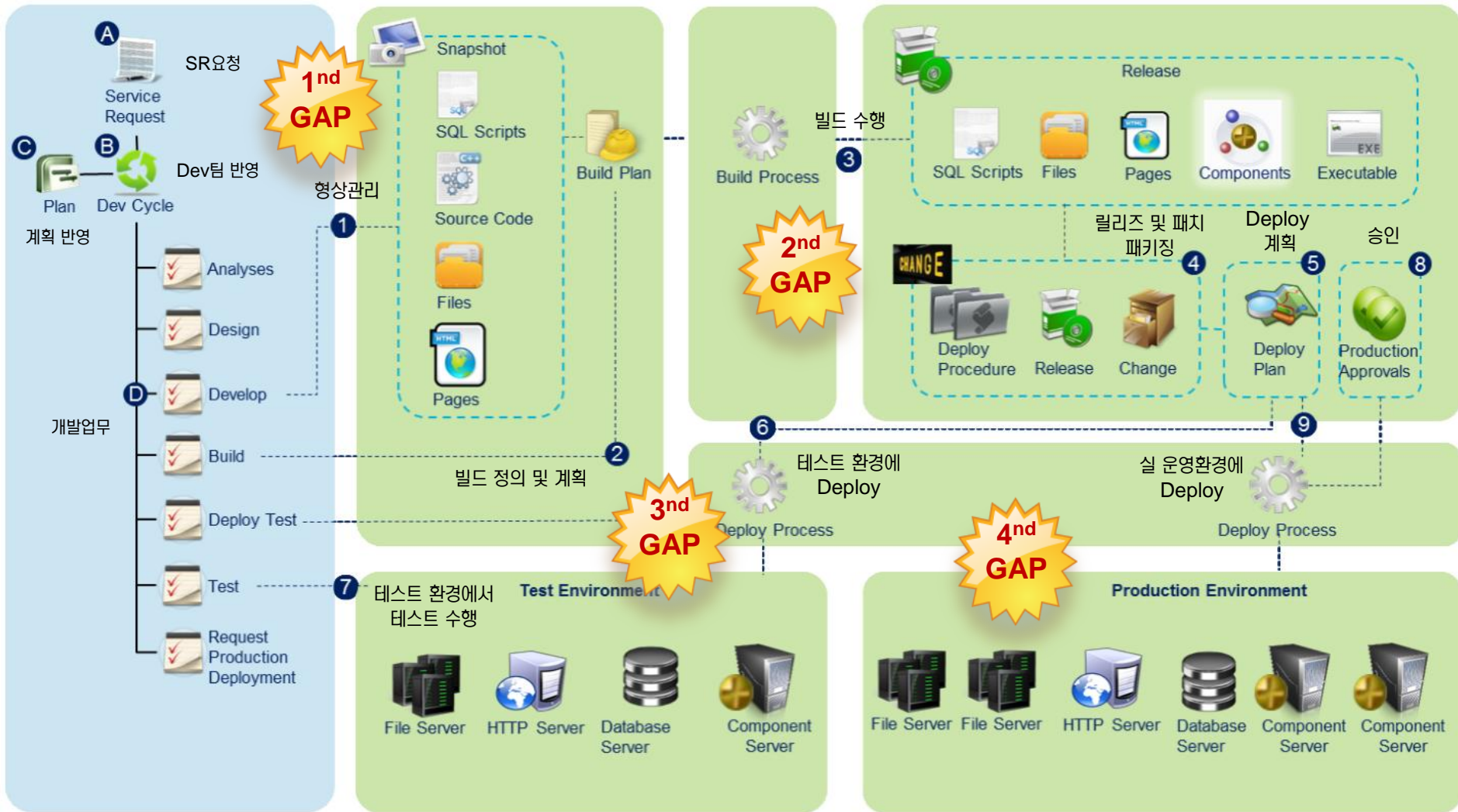
적은 비용으로 우수한 품질의 서비스를 빠르고 안정적으로 운영하기 위해서는 **저해요소가 무엇인지의 통찰과 혁신**이 필요

금융 IT 인프라의 규모와 복잡도는 점차 높아지고 있는 현실

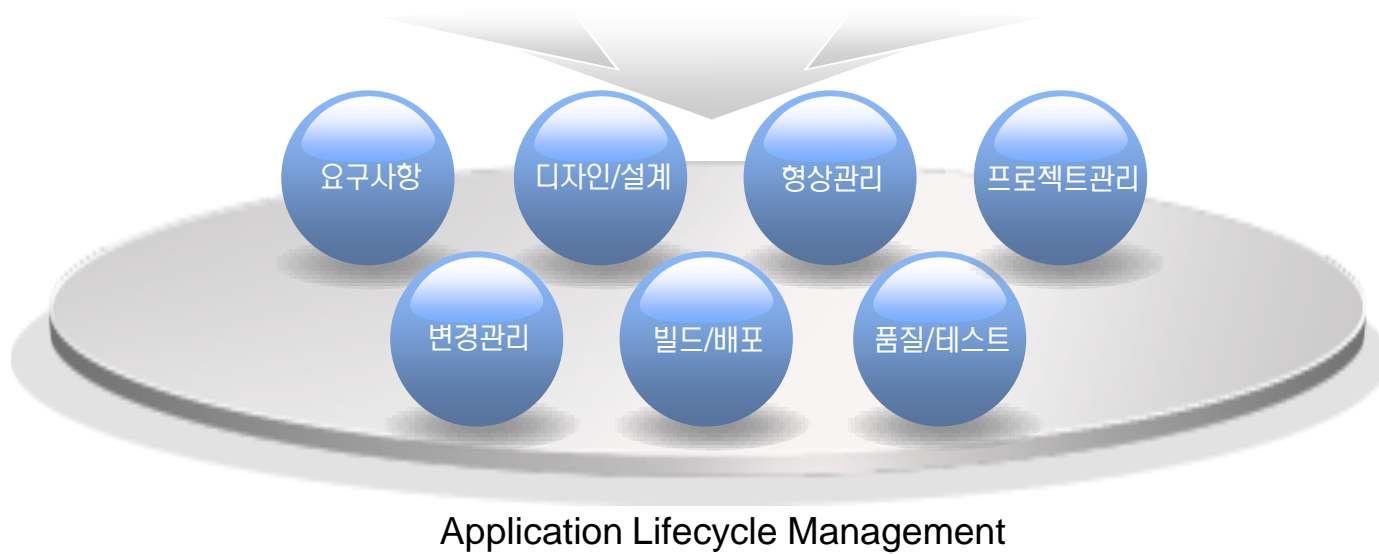
오늘날의 비즈니스와 기술적 요구는 전통적 방법만으로 운영하기엔 한계점에 이른 상황



Do More with Less가 더욱 어려워진 환경



현실: 개발 라이프 사이클 전반에 GAP 존재



회고 : ALM을 통한 해결 노력 (2005년 ~)



개발(Development)



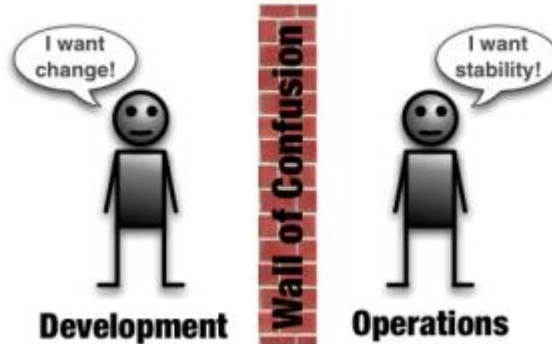
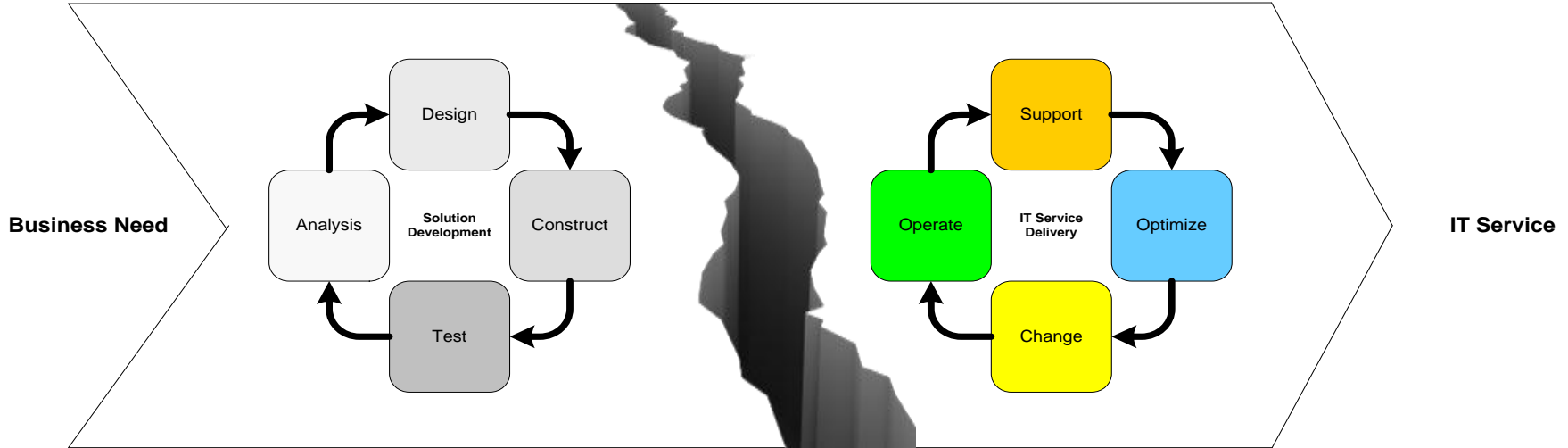
운영(Operations)

하지만, 현실의 가장 큰 장벽 ... 개발&운영간 불신

빠른 개발과 배포, 신기술

상반된 동기

안정화, 품질



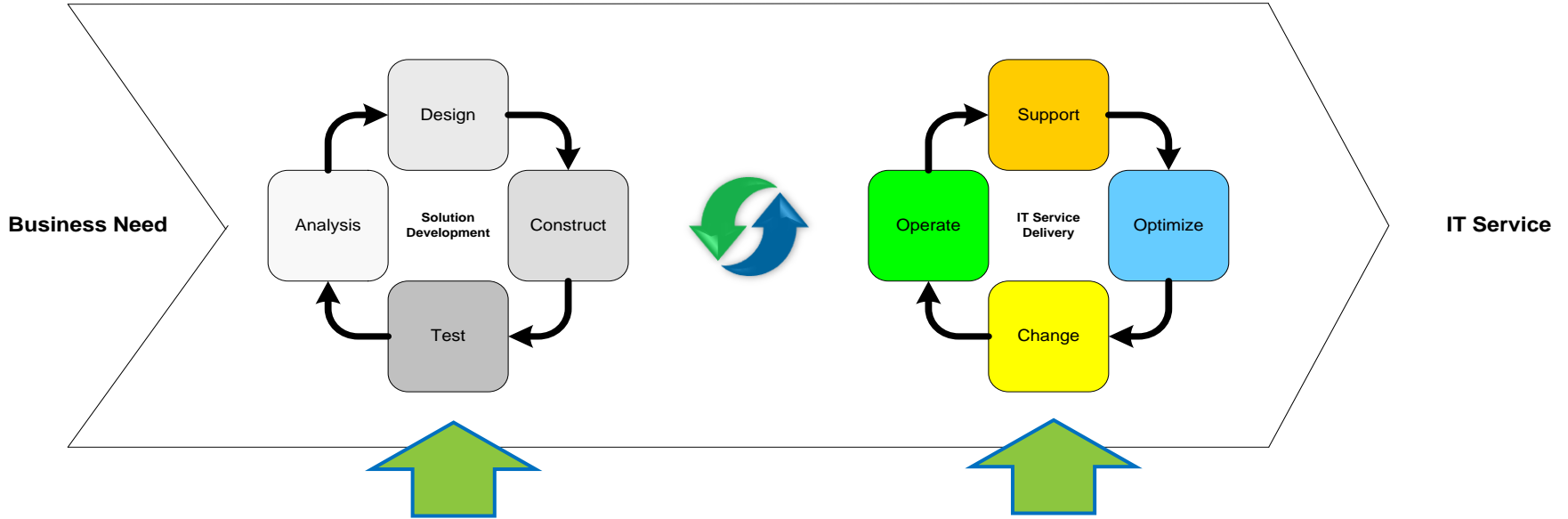
Dev & Ops간 이슈들

- 서로 다른 언어/용어 사용
- 고립 된 데이터
- 연계되지 않는 업무흐름
- 연계가 힘든 툴들 사용

현 조직간의 부조화 이유: 상반된 동기

빠른 개발과 배포, 신기술

안정화, 품질



해결방안:

*품질을 보다 높은
빠른 개발과 배포*

*보다 빠른 품질검토와 피드백을
통한 안정화와 품질관리*

상반된 동기 속에서 해결 방안 찾기

Agenda

2. DevOps의 등장과 개요

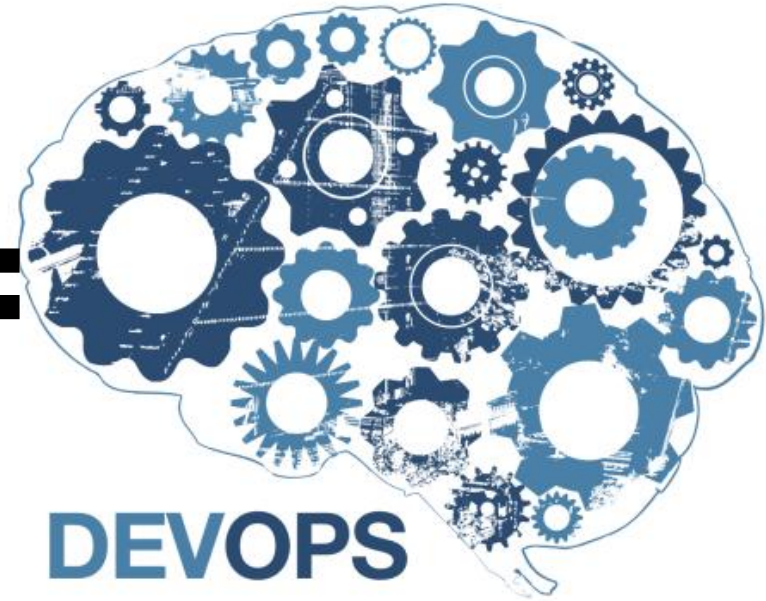
Contents

- DevOps의 등장
- DevOps의 개요
- DevOps를 통한 개선 및 혁신

Dev + Ops =

Development
개발

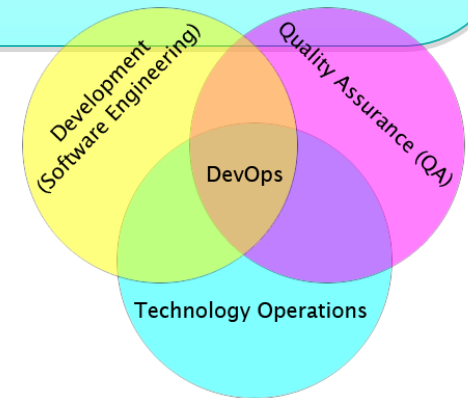
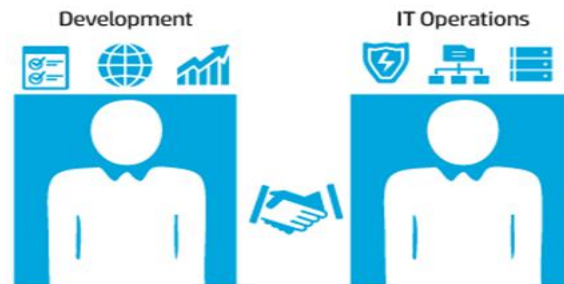
Operations
운영



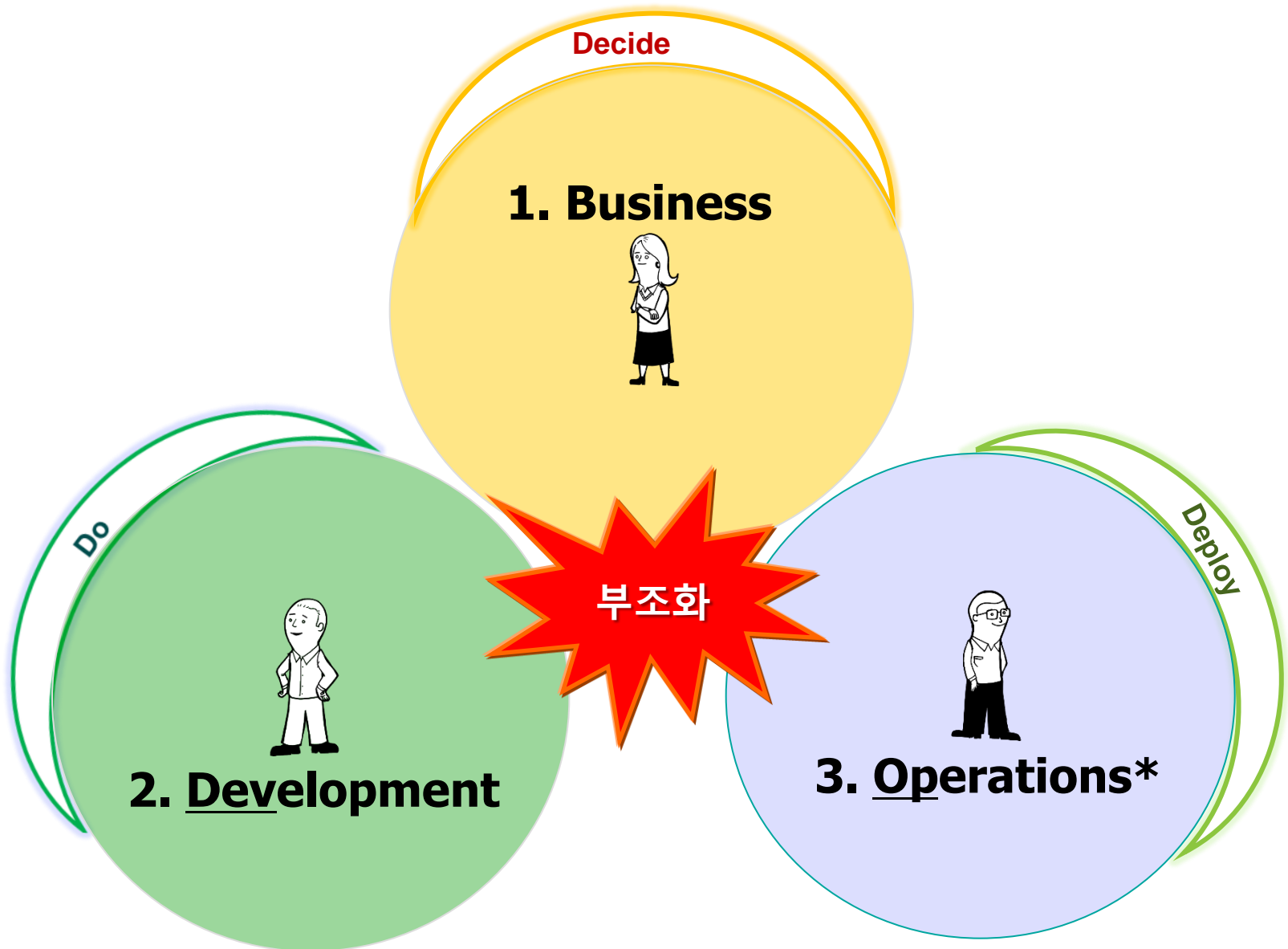
DEVOPS

Wikipedia's 정의

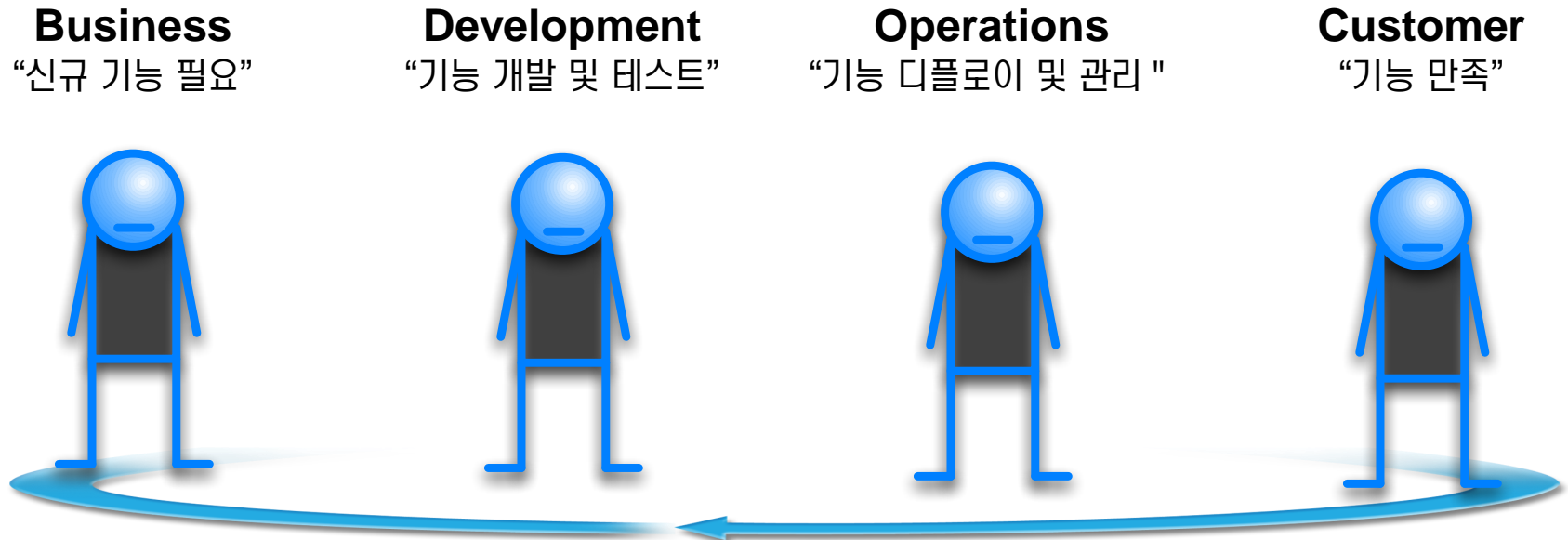
DevOps (개발과 운영의 합성어)은 소프트웨어 개발자와 IT운영전문가들 간의 커뮤니케이션, 협업 및 통합을 강조하는 소프트웨어 개발 방법론입니다. DevOps는 소프트웨어 개발과 IT 운영간의 상호의존 관계를 개선하기 위해 등장했고, 소프트웨어 제품 및 서비스를 빠르게 생산하려는 조직을 돕는 것을 목적으로 합니다.



What's DevOps?

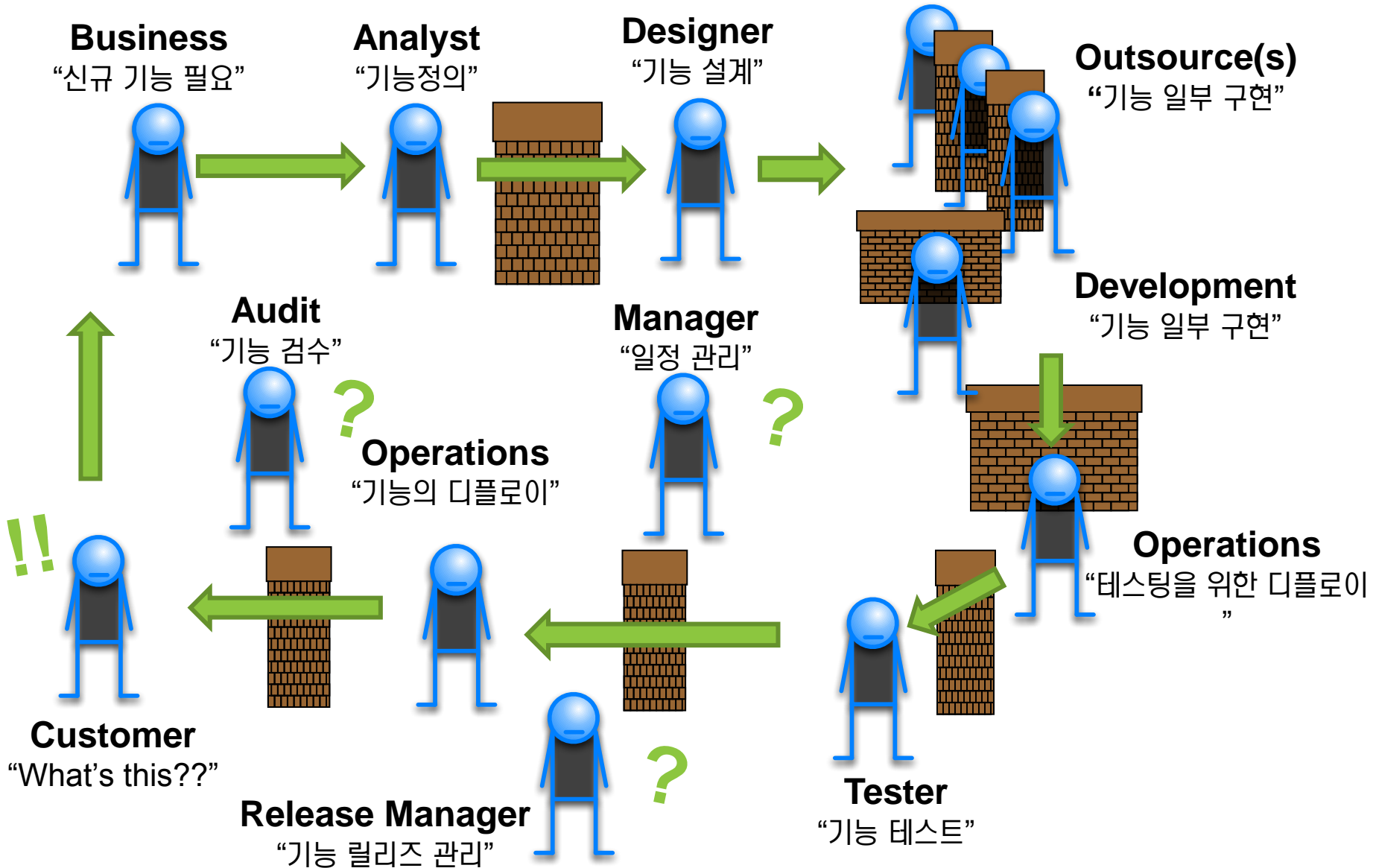


Software Delivery의 개념

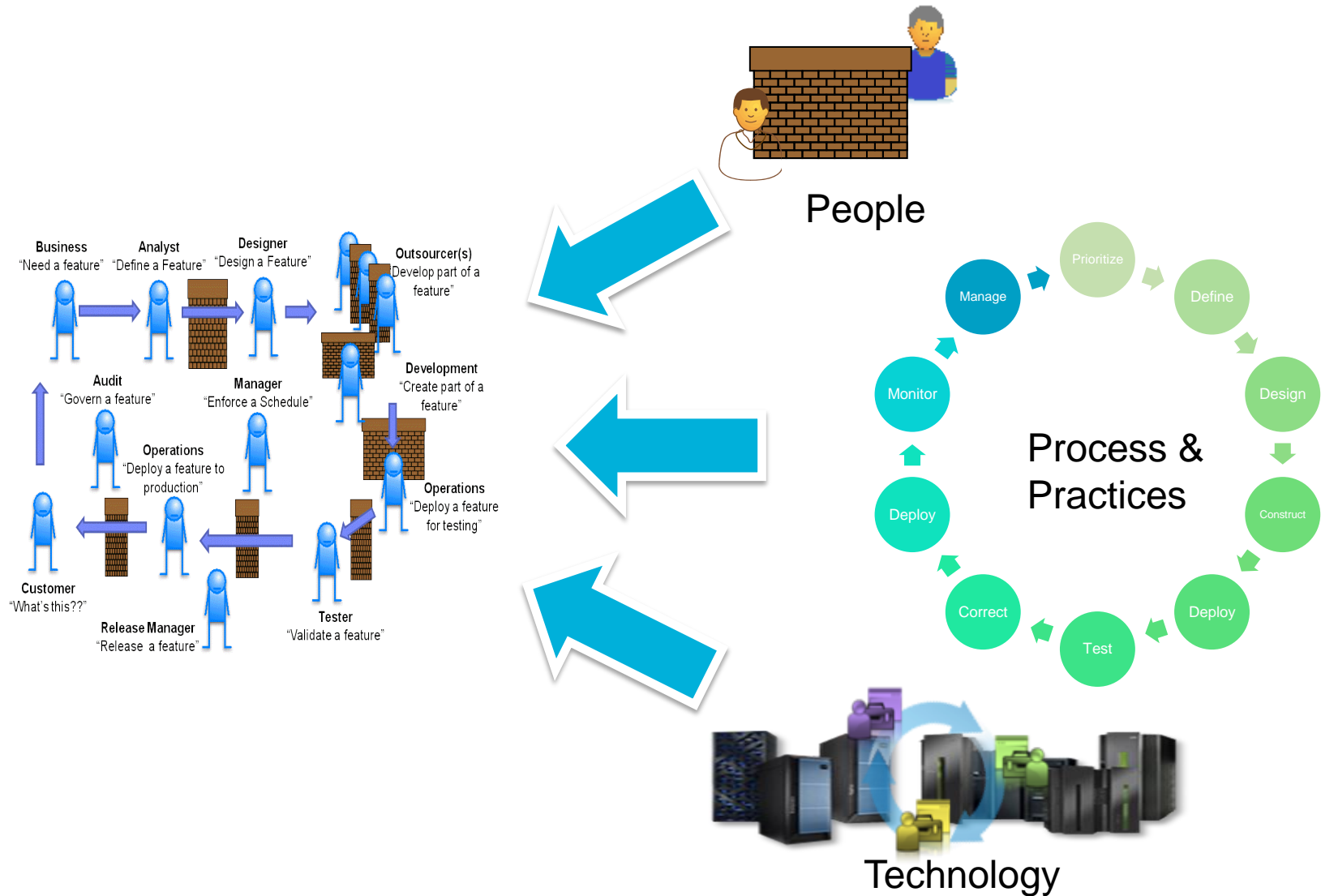


개념은 쉽고 간단하지만 ...

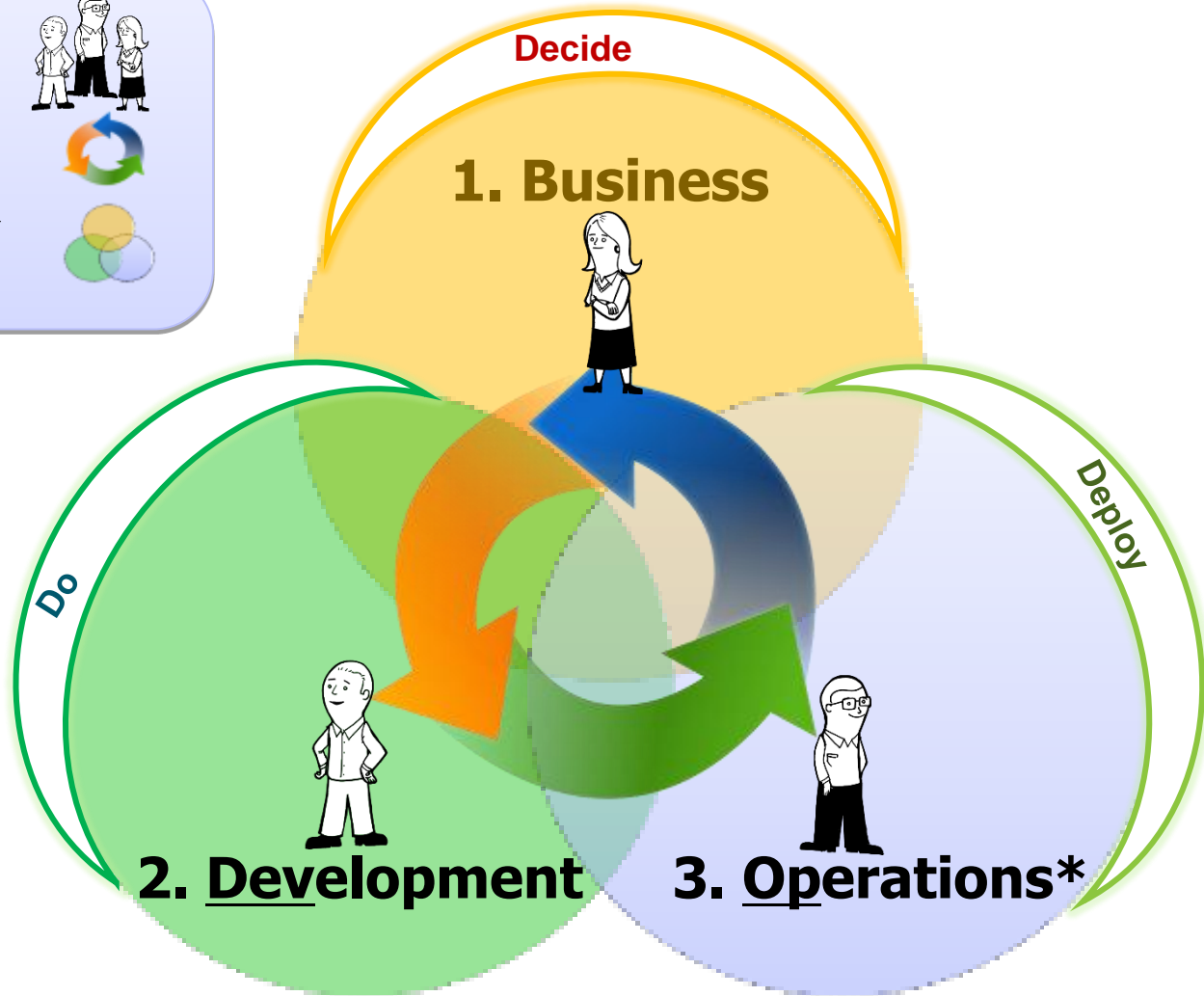




Software Delivery의 현실



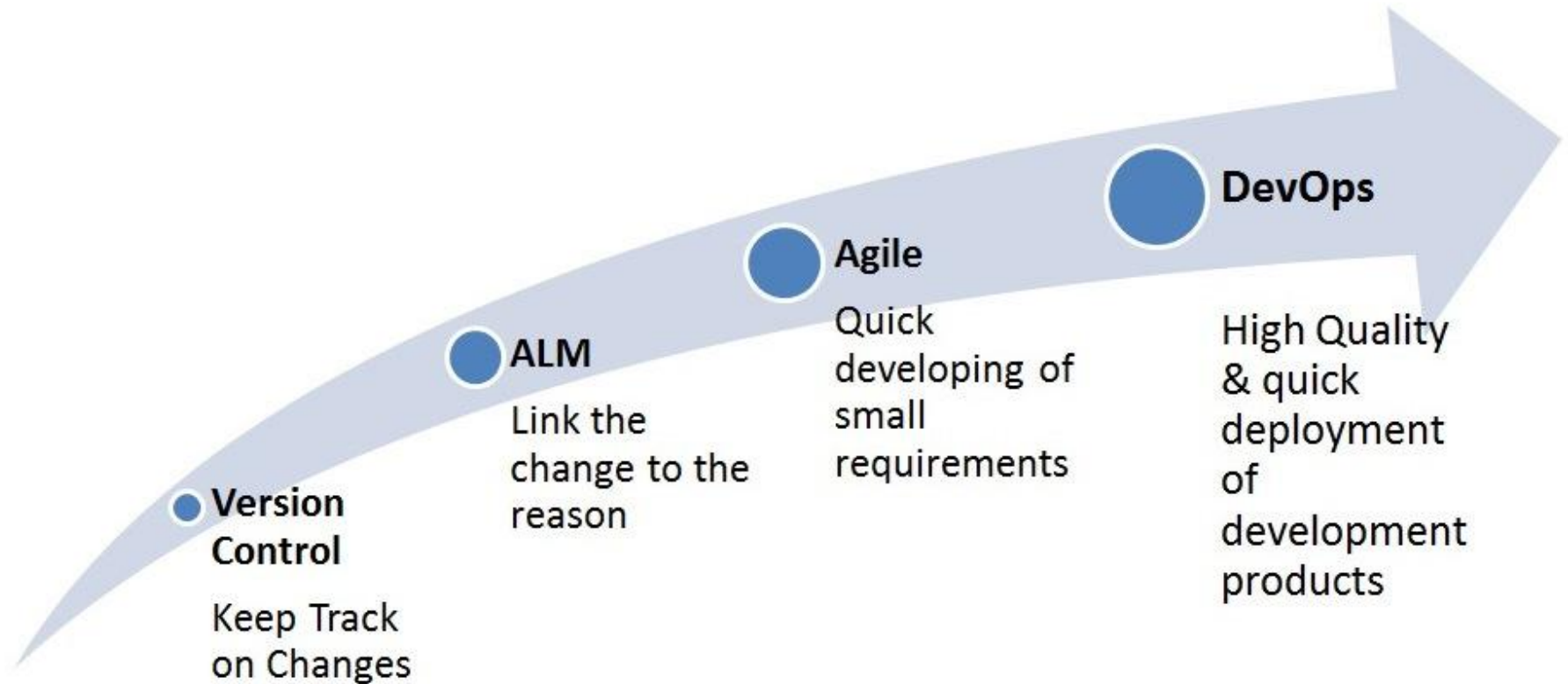
“People, Process, Technology”를 통한 Software Delivery 개선



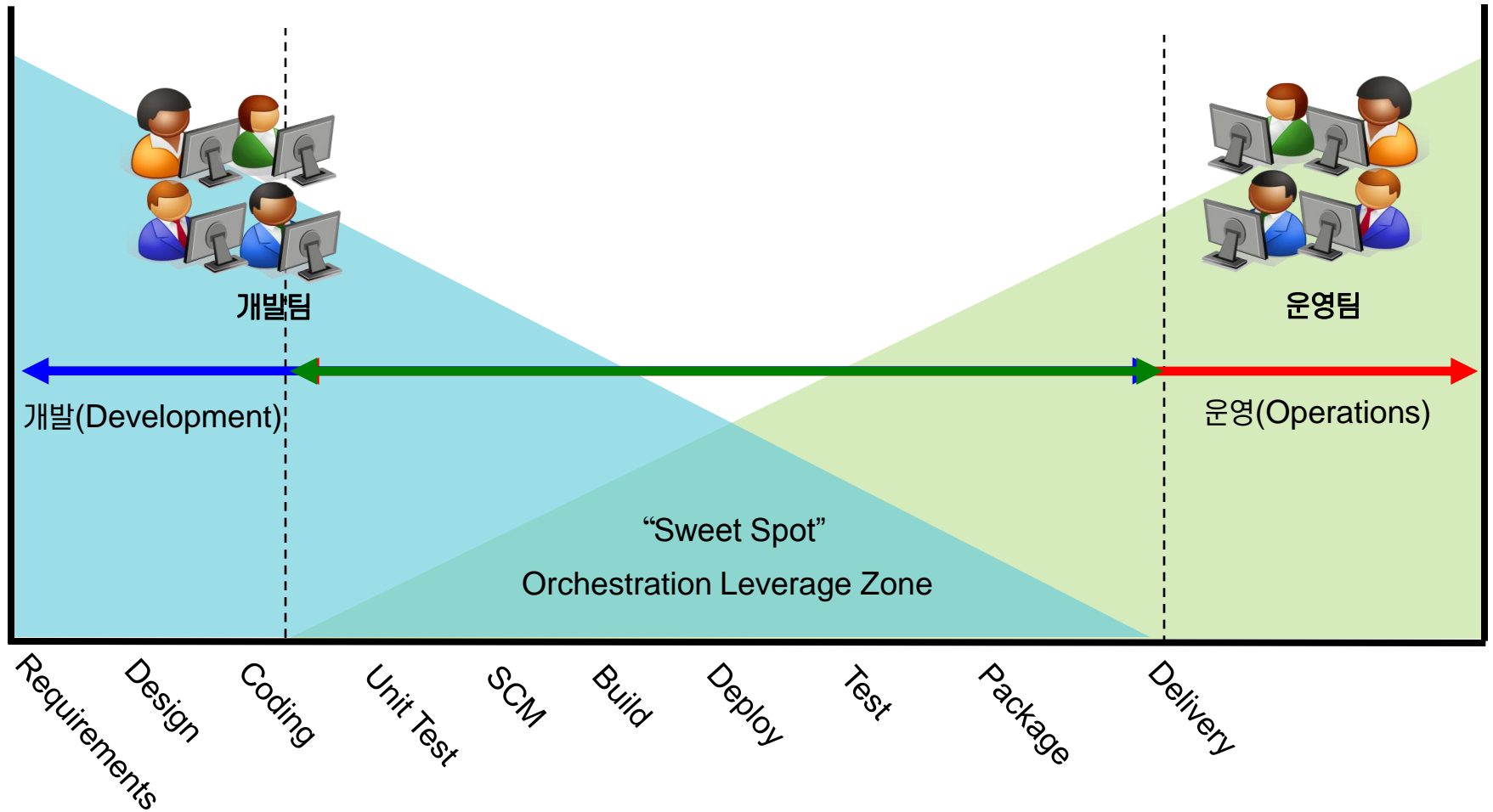
People
Process
Technology



방법론과 프로세스의 진화



IT 개발 방법/프로세스/도구의 진화



DevOps의 범위와 IT효율적 관리방안

“Sweet Spot의 재조명”

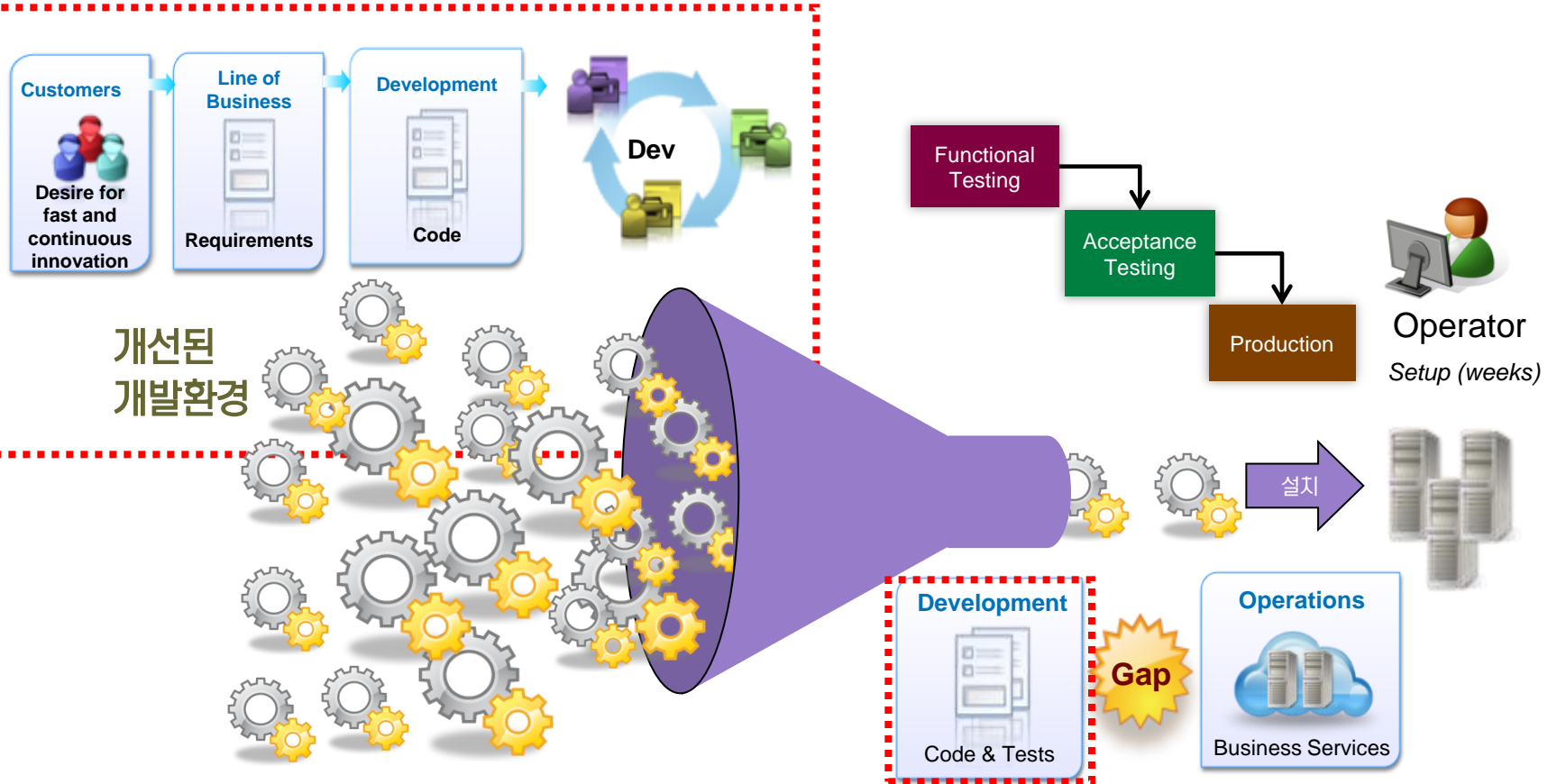
Agenda

3. DevOps를 통한 금융 IT 인프라 최적화 방안

Contents

- 병목 구간과 Gap에 대한 이해
- 병목과 Gap 해소 방안
- DevOps Best Practices의 이해

- 생산성과 품질 향상을 위해서는 개발/품질/운영상의 bottleneck 구간과 문제 이해가 중요합니다.



개발 환경의 개선을 통한 빠른 개발 진행.
 많은 개발이 이루어지지만, 테스트(품질)가 제대로 이뤄지지 않음.

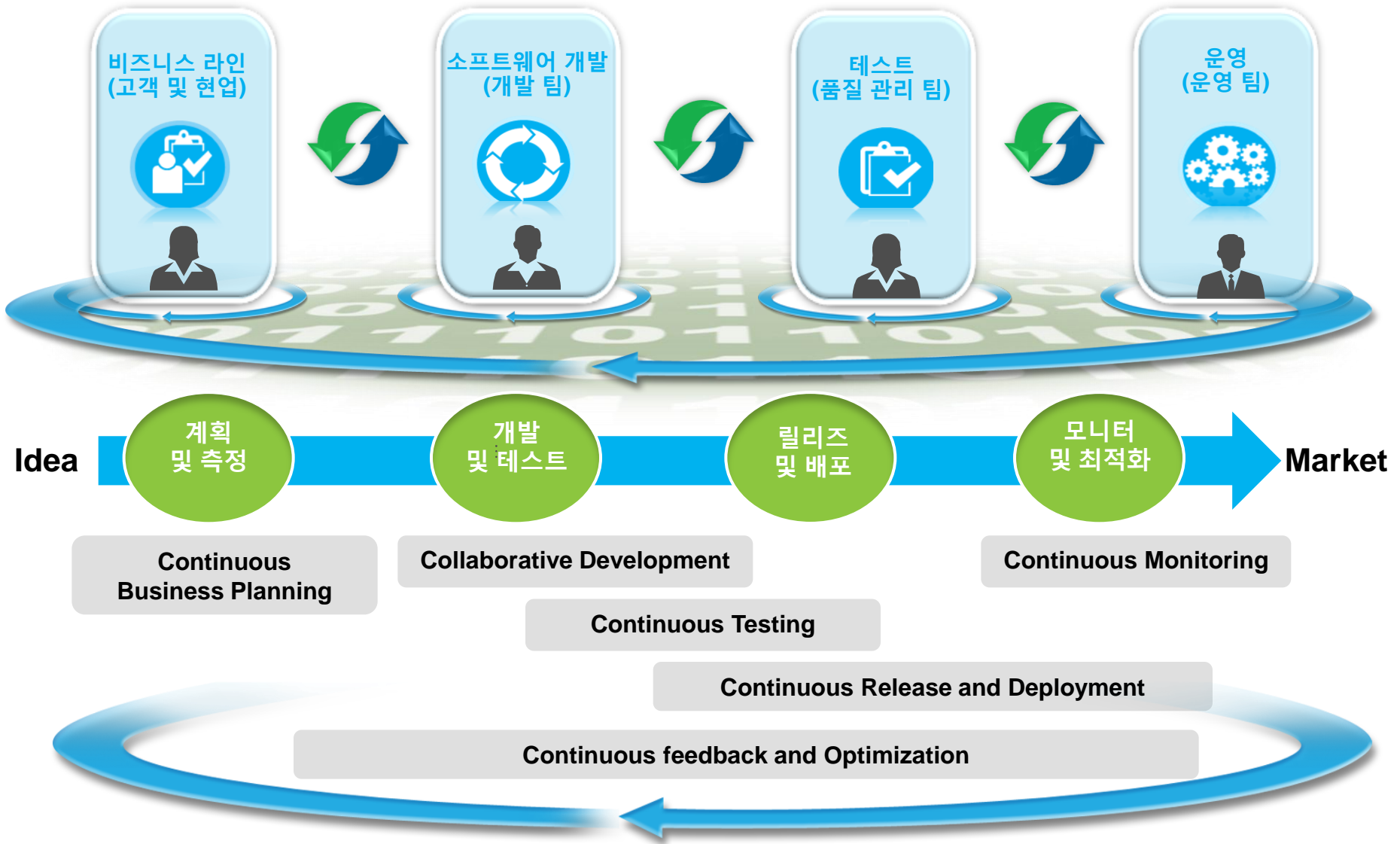
품질과 운영은 여전히 예전 방식을 유지하고
 있으면 이 영역에서 bottleneck 발생

병목(Bottleneck)과 Gap 해소를 통한 문제해결



기존의 노력 : 대부분 조직별 개선에 주력

금융 IT 인프라의 고도화 → “Gap의 해소”

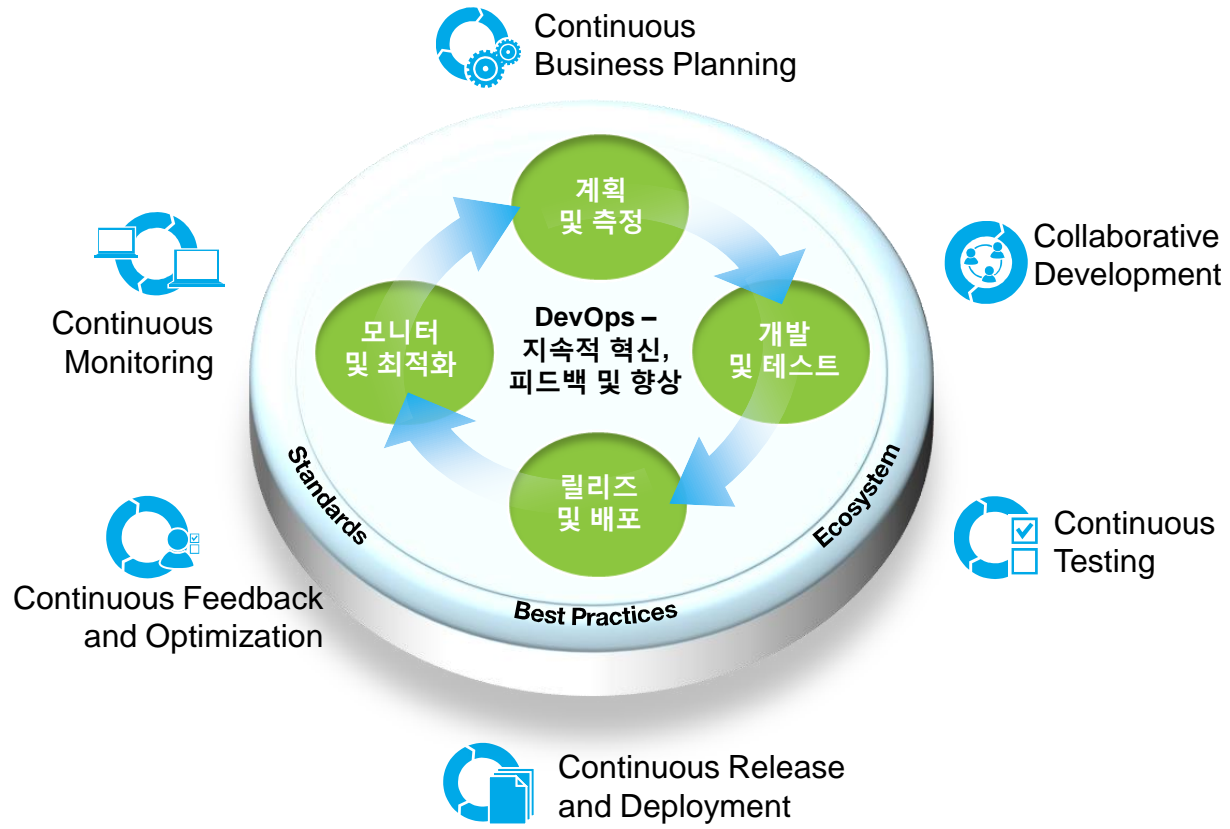


DevOps 대표 Practices

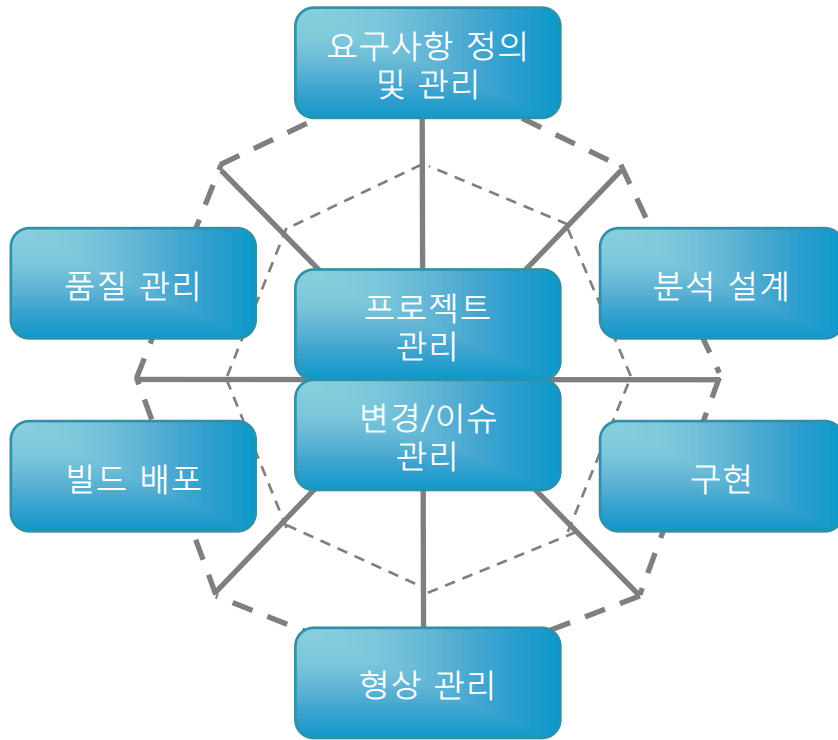
보다 빠른 *time to value*를 통한 :
소프트웨어 개발 가속화

혁신을 위해 추가 된 기능을 통한 :
속도, 비용, 품질, 위험요소간 밸런스

개선된 고객 경험을 통한 :
고객 피드백 시간 감소



DevOps 대표 Practices와 가치



ALM에서 DevOps 시대로의 변화 (2013~)



BEFORE

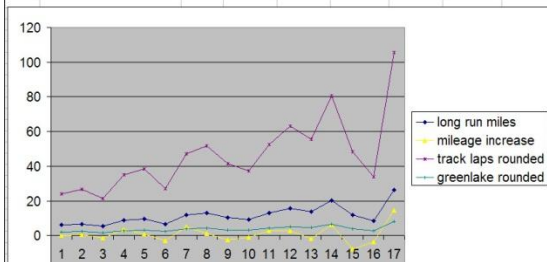


AFTER

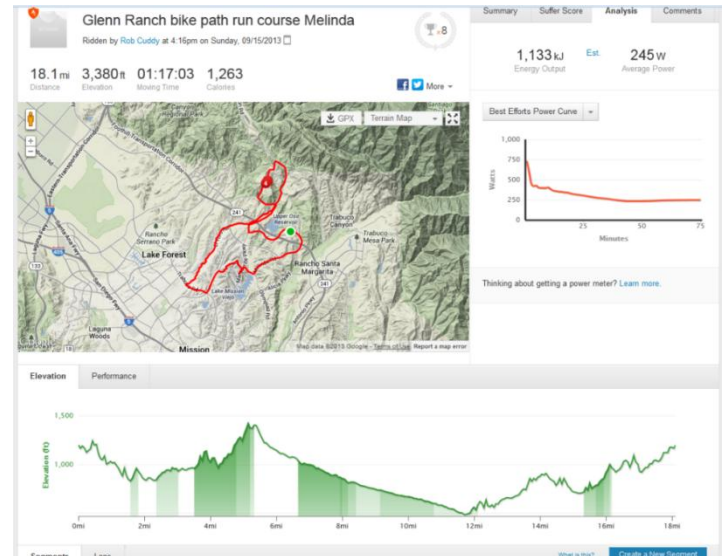
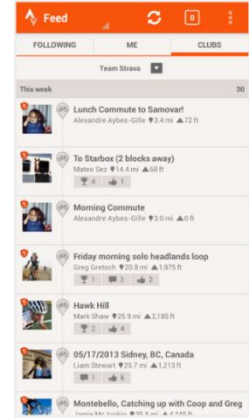
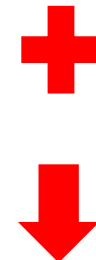
Usage1 : Continuous Business Planning



week #	dates	long run miles	% increase	mileage increase	laps around track	track laps rounded	laps around greenlake	greenlake rounded
1	Sunday, January 14, 2007	6	0.0%	0.00	24.0	24	1.9	2
2	Sunday, January 21, 2007	7	10.0%	0.60	26.4	27	2.1	2
3	Sunday, January 28, 2007	5	-20.0%	-1.32	21.1	21	1.7	2
4	Sunday, February 04, 2007	9	55.0%	3.43	34.8	35	2.7	3
5	Sunday, February 11, 2007	10	10.0%	0.87	38.3	39	3.0	3
6	Sunday, February 18, 2007	7	-30.0%	-2.87	26.8	27	2.1	2
7	Sunday, February 25, 2007	12	75.0%	5.03	47.0	47	3.7	4
8	Sunday, March 04, 2007	13	10.0%	1.17	51.7	52	4.0	4
9	Sunday, March 11, 2007	10	-20.0%	-2.58	41.3	42	3.2	3
10	Sunday, March 18, 2007	9	-10.0%	-1.03	37.2	37	2.9	3
11	Sunday, March 25, 2007	13	30.0%	2.79	52.6	53	4.1	4
12	Sunday, April 01, 2007	16	20.0%	2.62	63.0	63	4.9	5
13	Sunday, April 08, 2007	14	-12.0%	-1.89	55.4	56	4.3	5
14	Sunday, April 15, 2007	20	45.0%	6.23	80.4	81	6.3	7
15	Sunday, April 22, 2007	12	-40.0%	-8.04	49.2	49	3.8	4
16	Sunday, April 29, 2007	8	-30.0%	-3.62	33.8	34	2.6	3
17	Sunday, May 06, 2007	26	170.0%	14.34	105.6	106	8.2	8



BEFORE

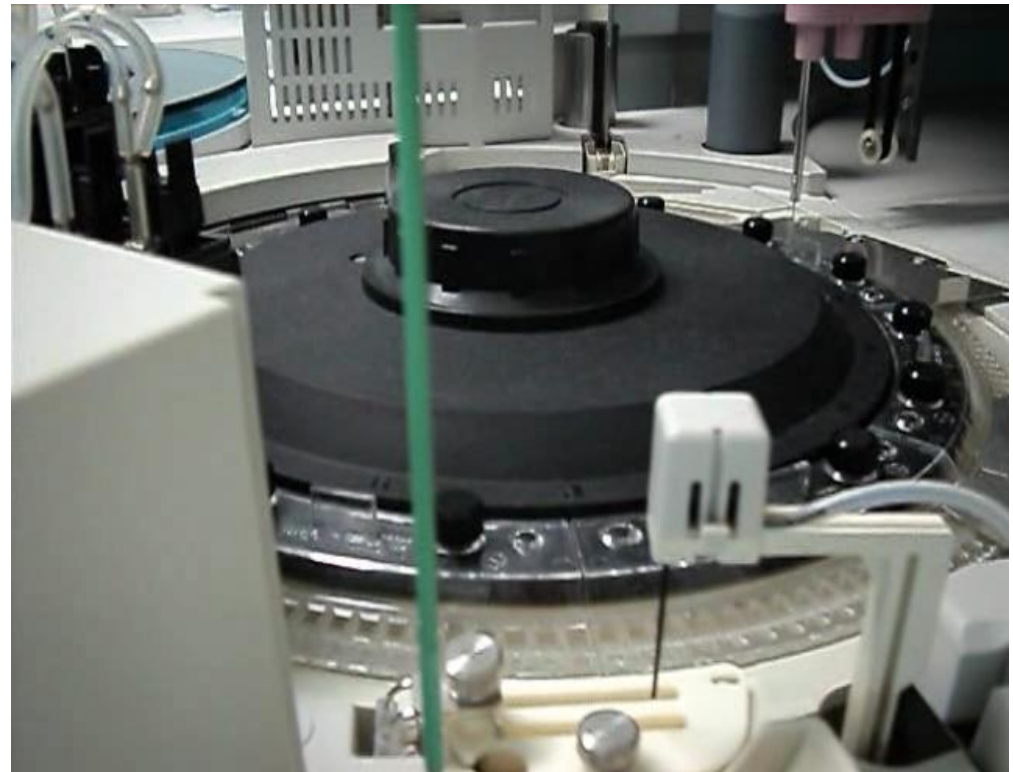


AFTER

Usage2 : Collaborative Development



BEFORE



AFTER

Usage3 : Continuous Testing



Usage4 : service virtualization



BEFORE



AFTER

Usage5 : Continuous Delivery

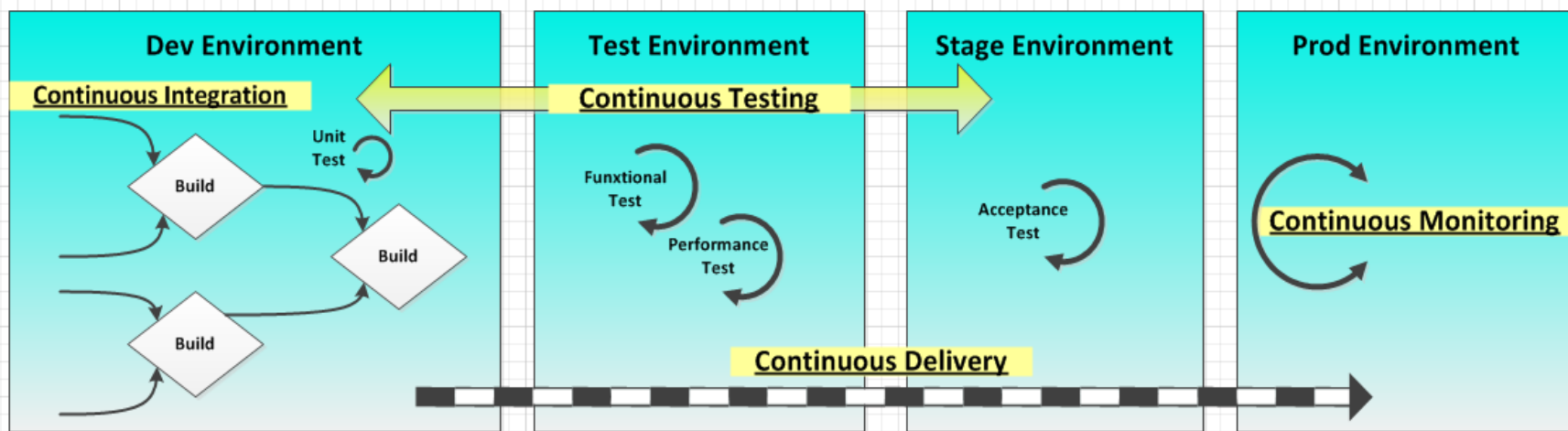
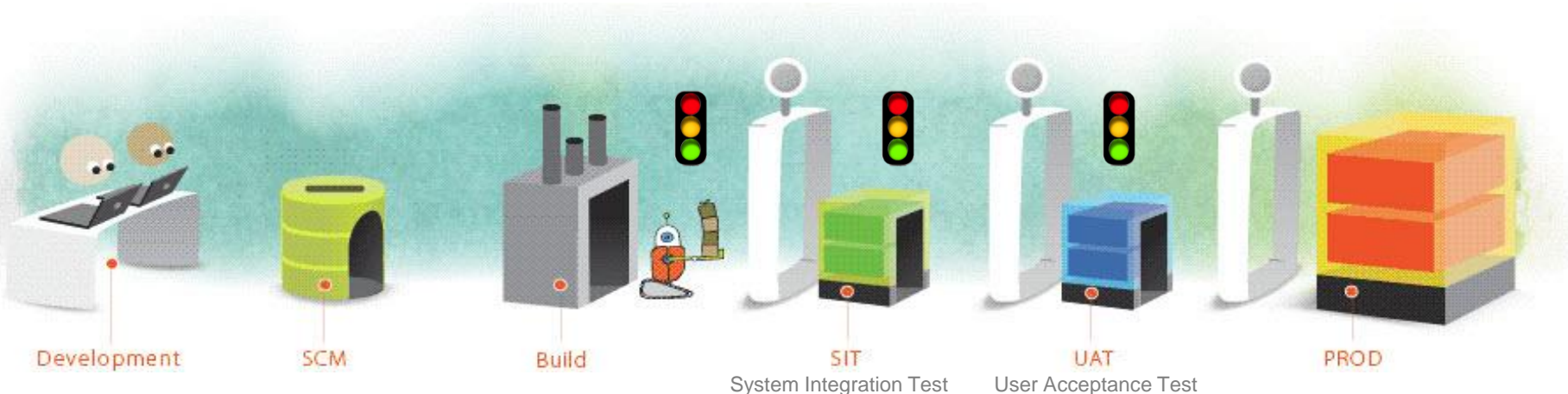
Agenda

4. DevOps기반 금융 IT 인프라 최적화 방안

Contents

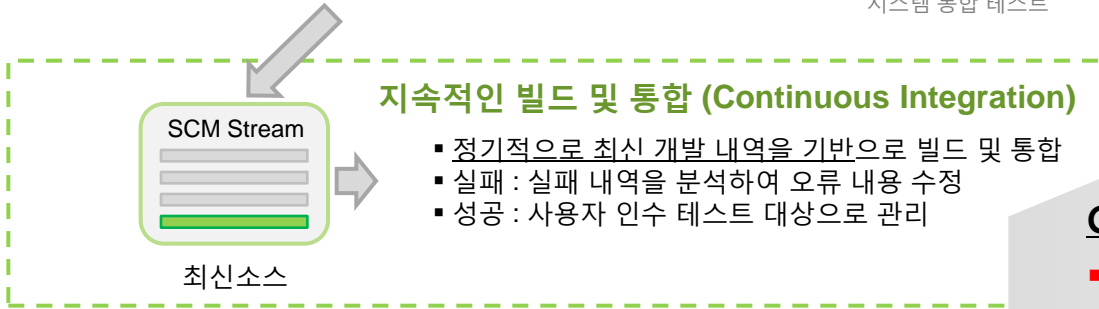
- 2014년 주목해야 할 포인트
- CI, CT, CD를 이용한 개선방향
- DevOps기반 최적화 방안

2014년, DevOps관점의 주목해야 할 포인트



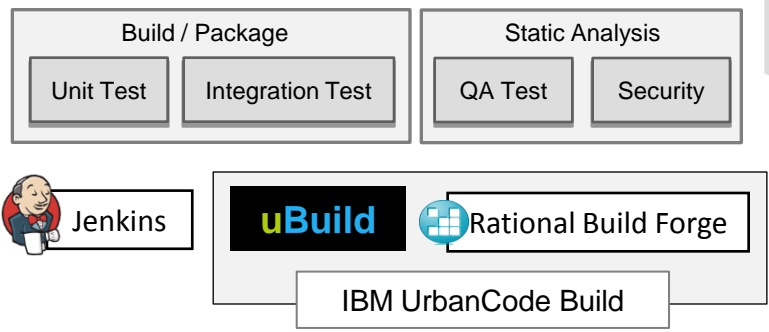
생산성 향상과 품질 향상을 위해서는 CI를 넘어 CD까지 고려

Continuous Integration(CI)를 이용한 개선



CI 만으론 문제 해결의 한계

- 빌드 이후, 바이너리 파일의 관리 이슈
- QA 프로세스 및 도구와의 연계 이슈
- 많은 공수의 커스터마이제이션



Continuous Testing(CT)을 이용한 개선



지속적인 빌드 및 통합 (Continuous Integration)

SCM Stream

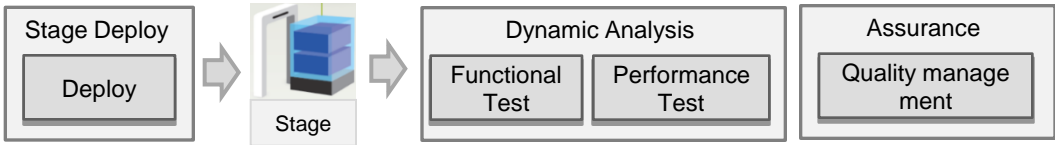
- 정기적으로 최신 개발 내역을 기반으로 빌드 및 통합
- 실패: 실패 내역을 분석하여 오류 내용 수정
- 성공: 사용자 인수 테스트 대상으로 관리

지속적인 테스트 (Continuous Testing)

SCM Stream

- 빌드/통합이 성공적으로 이뤄진 소스의 테스트 서버로의 배포
- 테스트 전문 도구를 이용한 테스트
- Defect 발생 시, 처리 프로세스에 따른 문제 해결
- 기능 테스트, 성능 테스트 / Black-Box, White-Box 테스트

- Rational Quality Manager
- Rational Test Workbench
- Rational Test Virtualization Server



Continuous Release & Deploy 를 이용한 개선



지속적인 빌드 및 통합 (Continuous Integration)

SCM Stream

- 정기적으로 최신 개발 내역을 기반으로 빌드 및 통합
- 실패 : 실패 내역을 분석하여 오류 내용 수정
- 성공 : 사용자 인수 테스트 대상으로 관리

지속적인 테스트 (Continuous Testing)

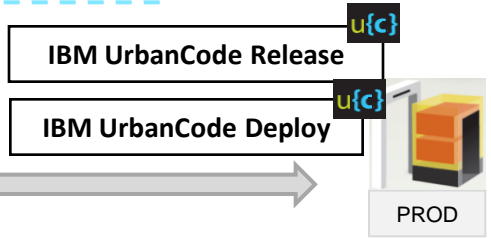
SCM Stream

- 빌드/통합이 성공적으로 이뤄진 소스의 테스트 서버로의 배포
- 테스트 전문 도구를 이용한 테스트
- Defect 발생 시, 처리 프로세스에 따른 문제 해결
- 기능 테스트, 성능 테스트 / Black-Box, White-Box 테스트

지속적인 릴리즈 & 디플로이 (Continuous Release & Deploy)

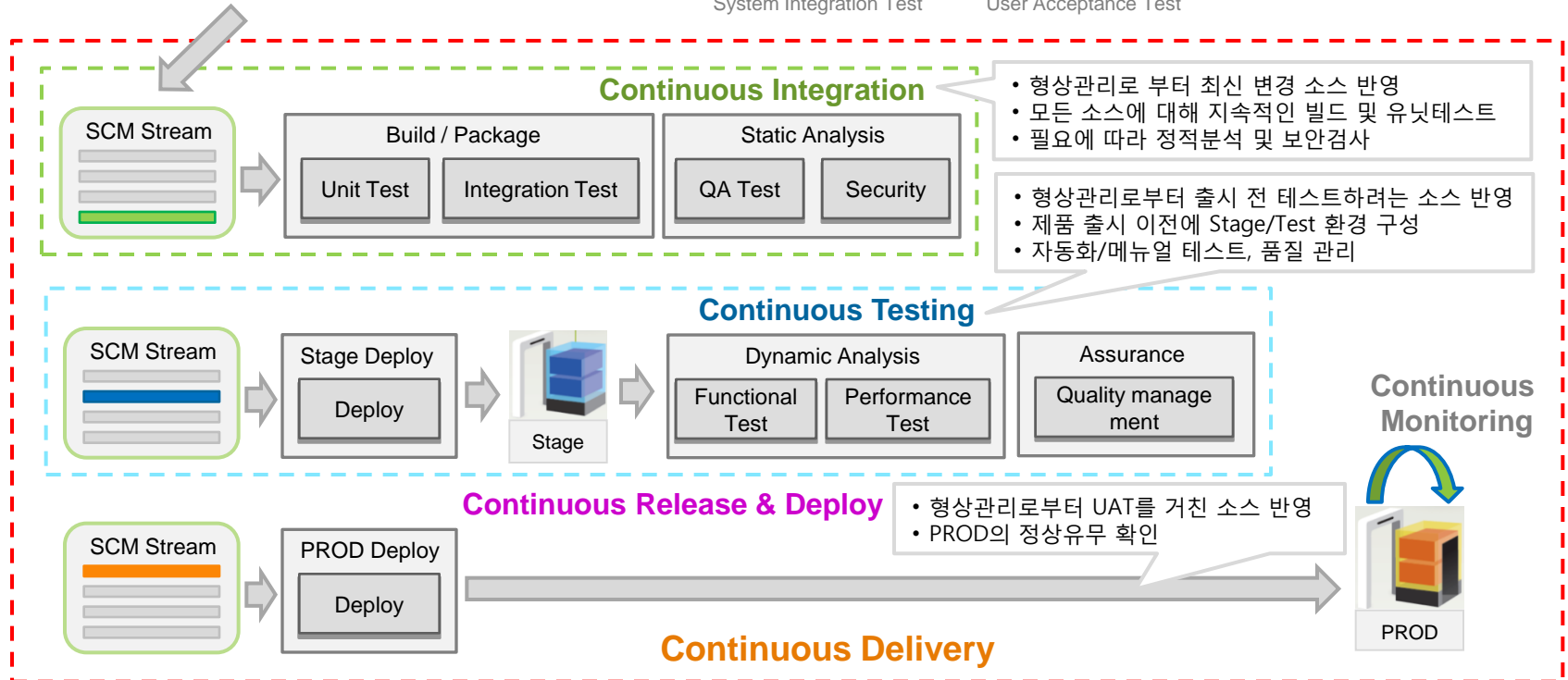
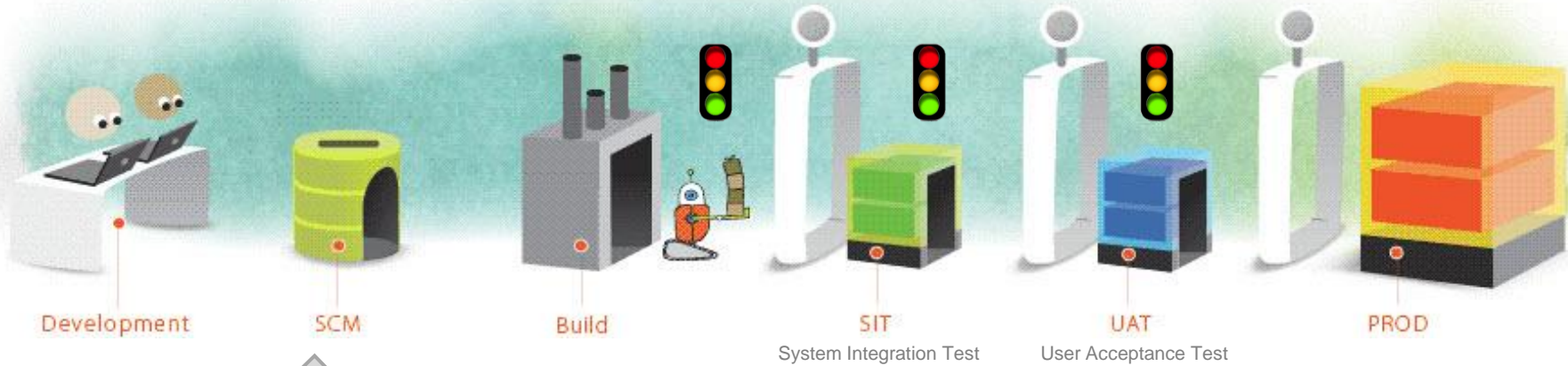
SCM Stream

- 사용자 인수 테스트까지 완료된 소스의 관리
- 테스트 완료된 소스기반으로 패키지 구성 및 승인
- 승인 절차를 마친 패키지 배포의 스케줄 배포

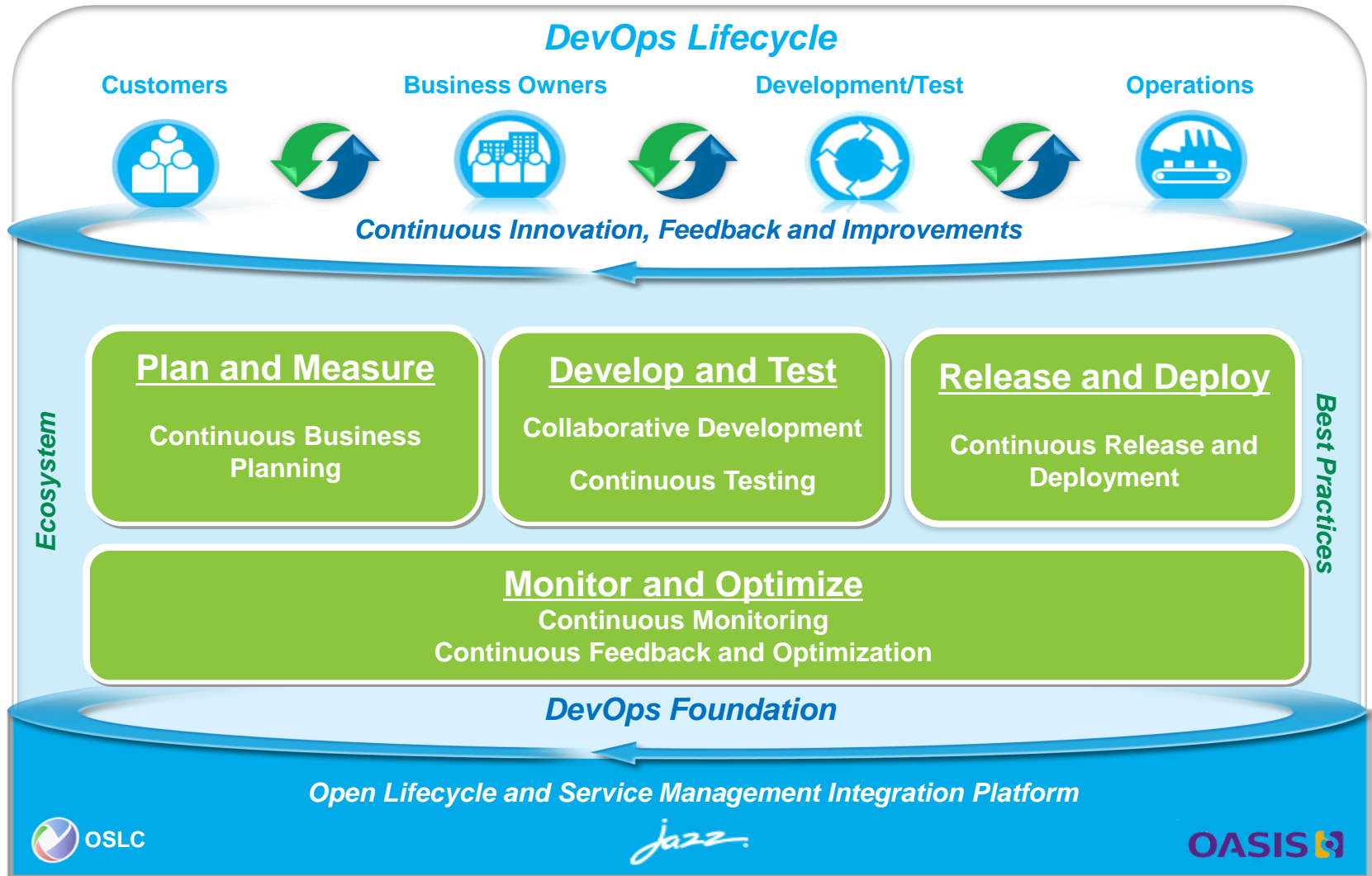


개발/테스트/운영 배포관리를 위한 릴리즈/디플로이 전문 솔루션

생산성 및 품질 향상을 위한 Continuous Delivery (CD)



DevOps기반의 금융IT 인프라 최적화 방안



QUESTIONS



Thank You

