

누구나 하는 DevOps

SpringCamp 2017

IBM Korea KONG, Jin gi
Bluemix Technical Evangelist

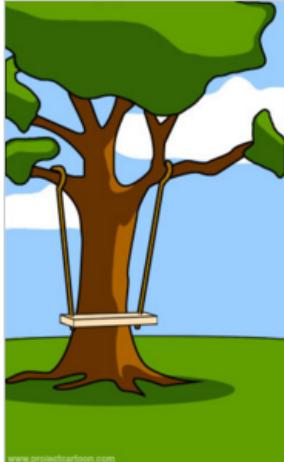


DevOps 란?

- Development + Operations ?
- 개발 및 운영 문화의 변경
- 빠른 개발, feedback 및 적용
- 자동화를 통해
 - 품질, 속도, 효율성 ↑
 - 비용, 오류 ↓
- 하지만..
 - 개발, 시스템과 아키텍처에 대한 깊은 이해 필요
 - (고오오급 개발자)
 - 높은 구축 비용
 - 유지보수 문제



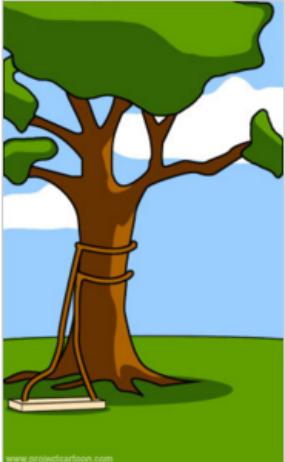
How the customer
explained it



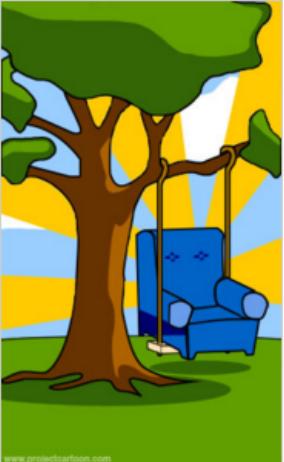
How the project
leader understood it



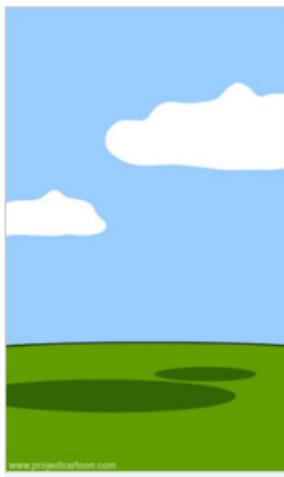
How the analyst
designed it



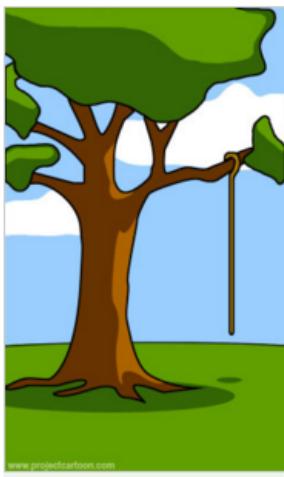
How the
programmer wrote it



How the business
consultant described
it



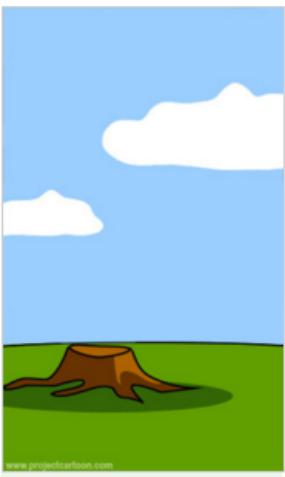
How the project was
documented



What operations
installed



How the customer
was billed



How it was
supported

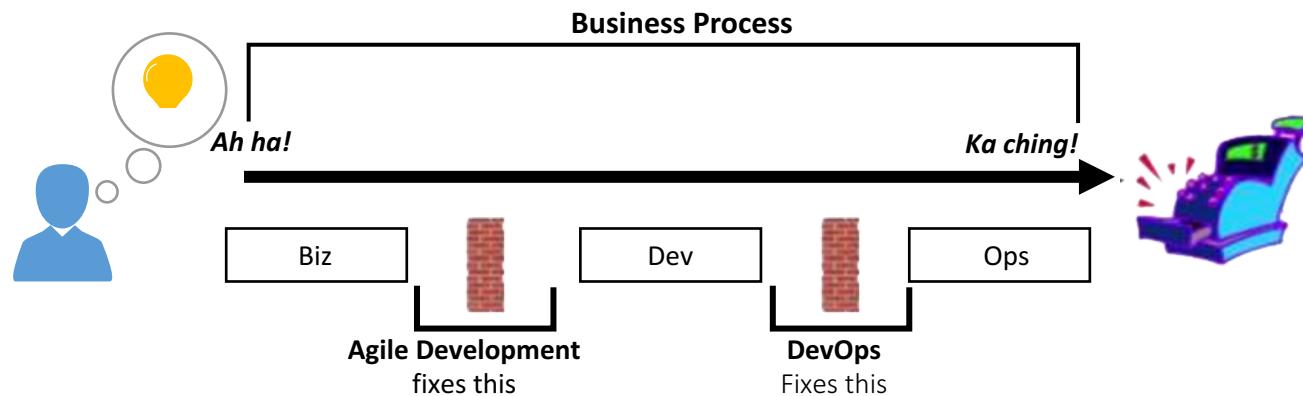


What the customer
really needed

<http://projectcartoon.com/>

애자일과 DevOps

- 아이디어를 빨리 구현하고 확인하여 feedback 을 통한 빠른 사이클이 애자일
- 개발과 배포, 관리를 개선하는 프로세스가 DevOps



왜 DevOps 가 뜨는가?

환경과 기술의 변화

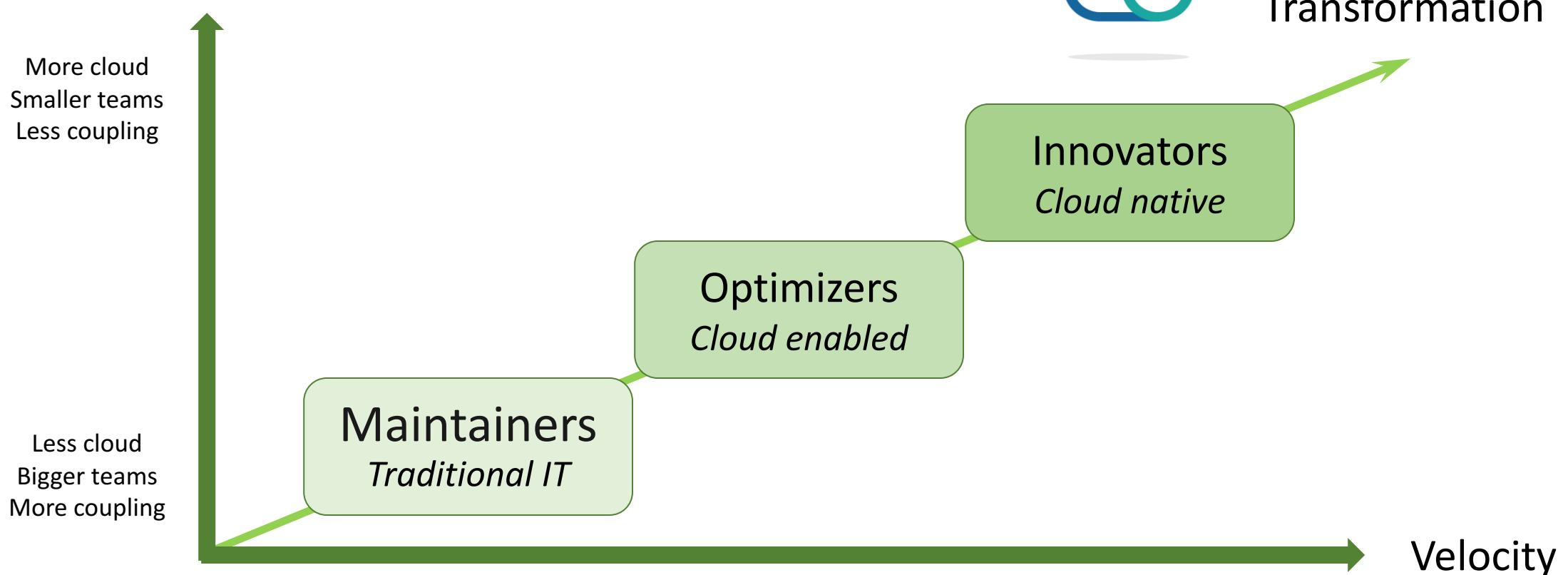
- 자동화 기술과 도구의 발전
- 클라우드 플랫폼의 확장
 - IaaS, PaaS, Container
- MicroService Architecture

시장과 경영의 요구

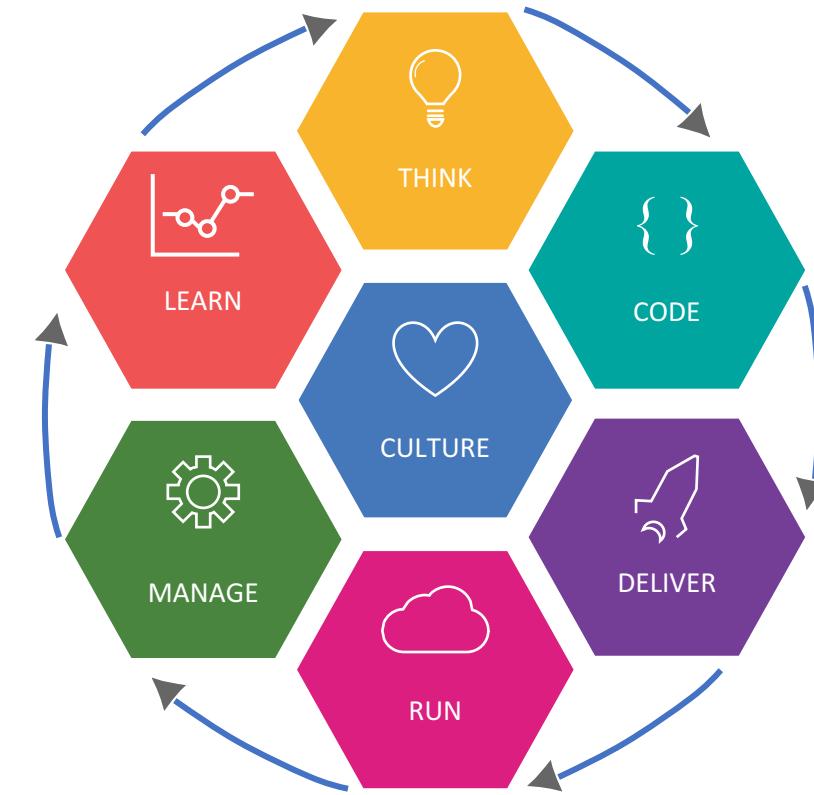
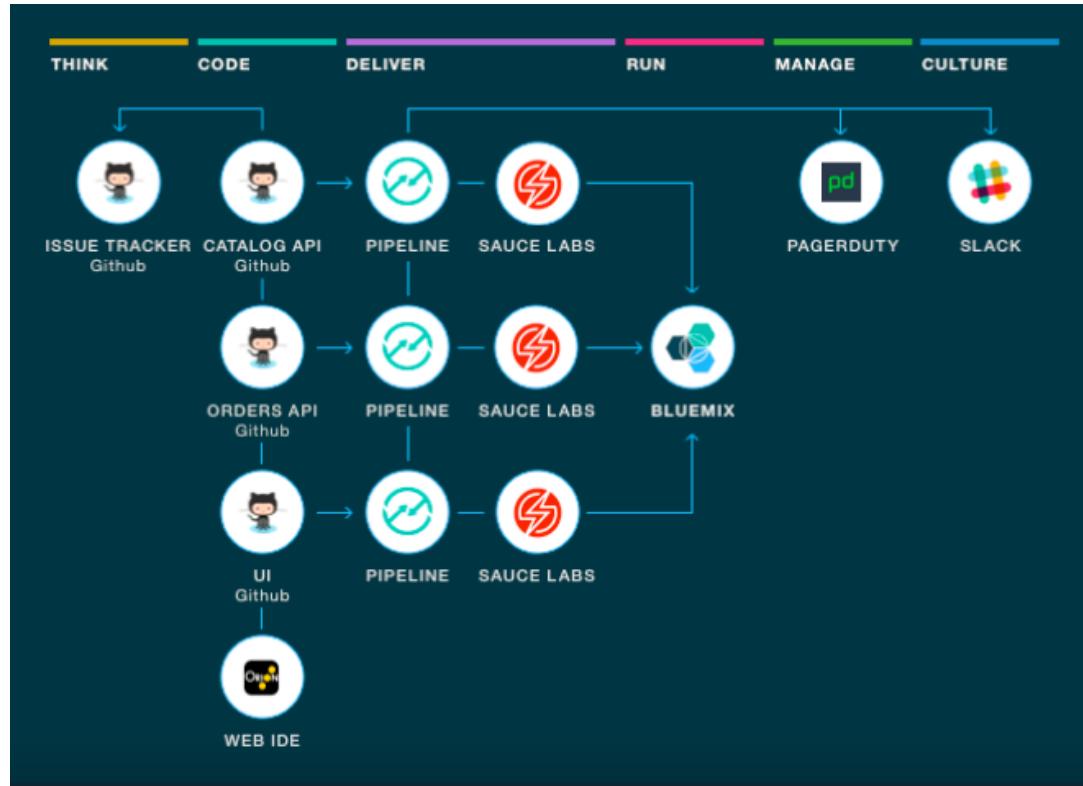
- 시장의 요구에 따른 빠른 개발과 변경
 - 요구에 맞는 기능
 - 빠른 기능 개선
 - 높은 완성도
 - 피드백 반영
- 비용 감소..?

DevOps 여정

Culture &
architecture

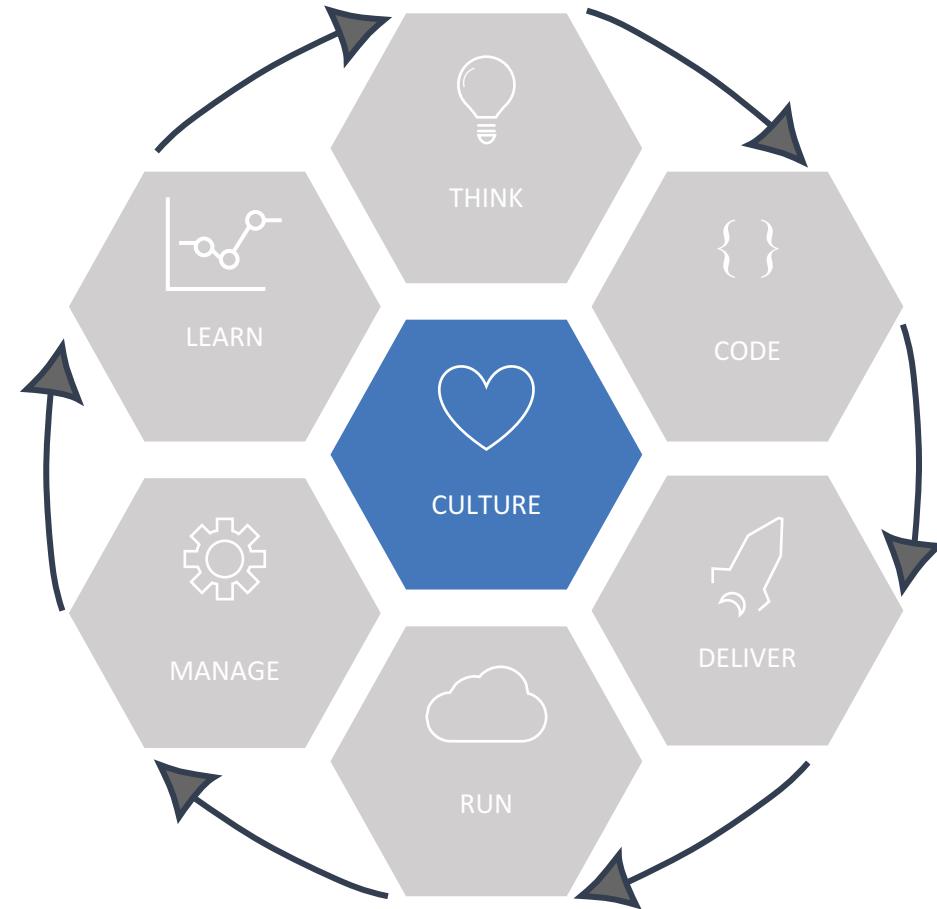


IBM Bluemix Garage Method

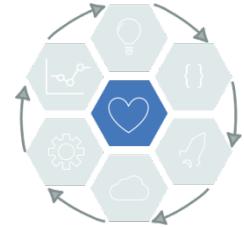


CULTURE

Transform & innovate with speed



새로운 환경을 위한 조직 문화의 변화



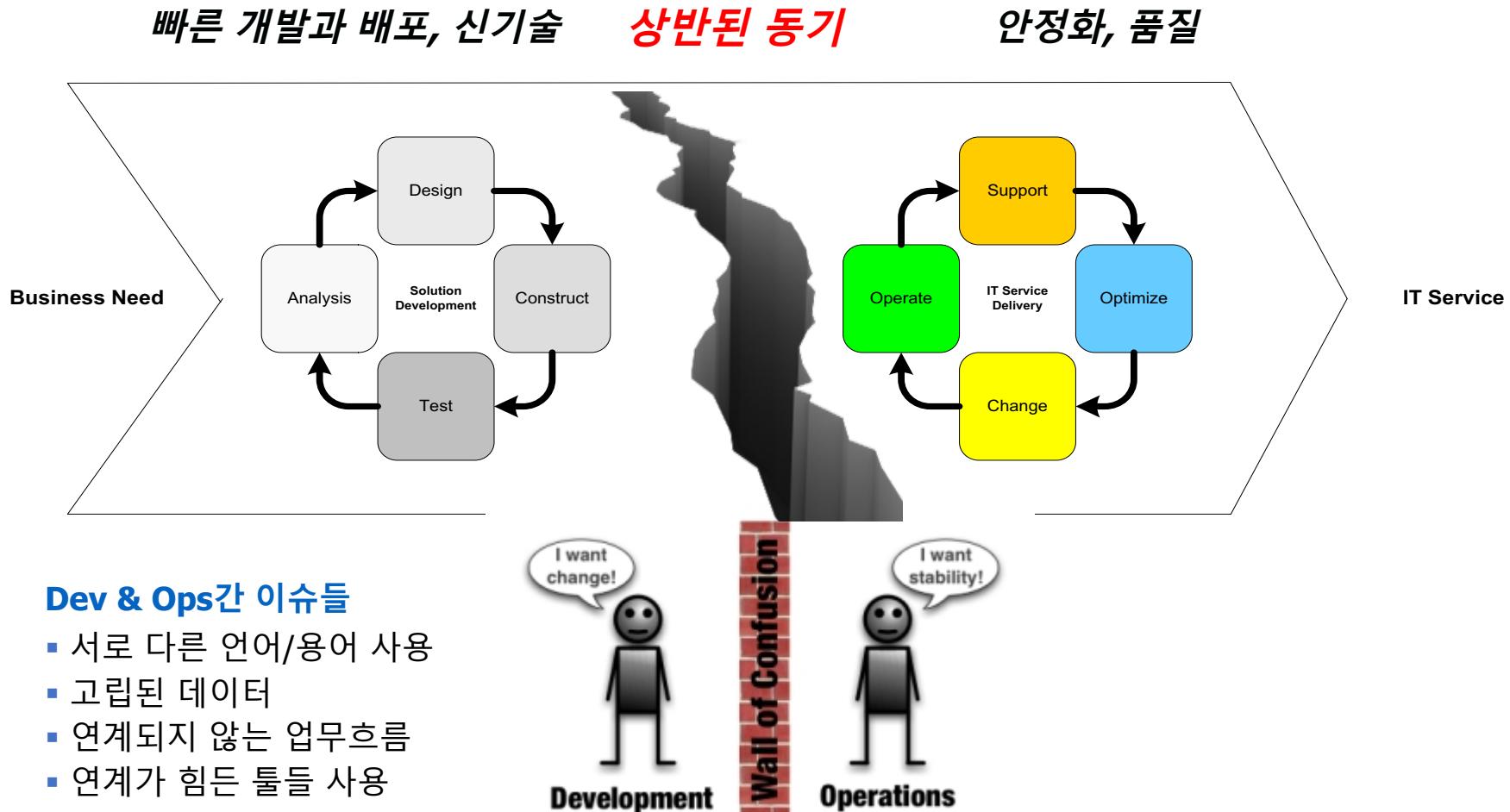
EVERYONE IN THE BOAT!

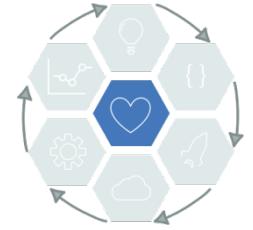
The business and product owner, designer, developer, and team lead all work together.

- 'Two Pizza Rule'
- 다양한 팀 구성
- 조직과 역할
- 같은 공간 내의 자율화된 팀
- 애자일 원칙에 기반

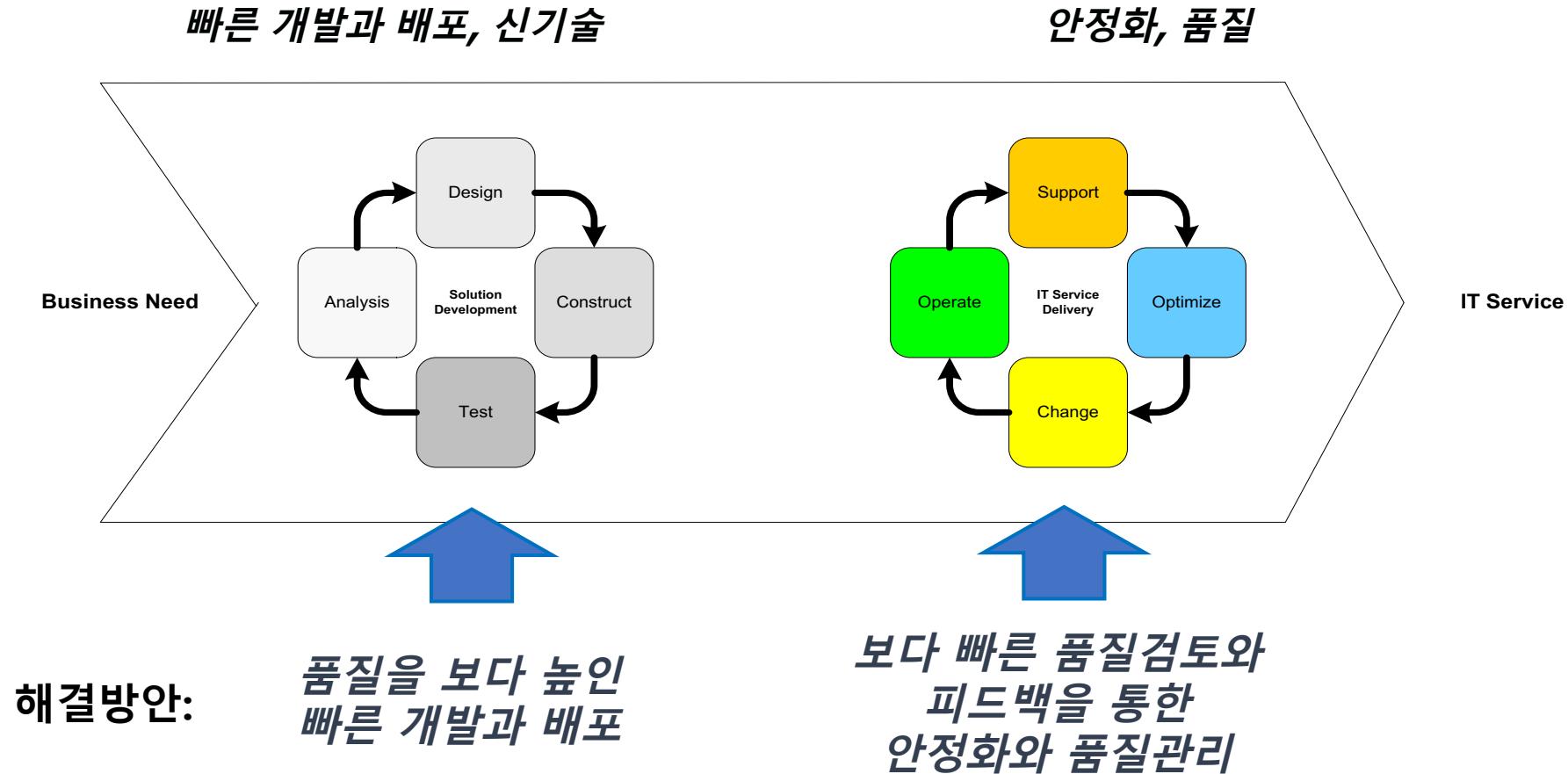


Developers vs. Operators

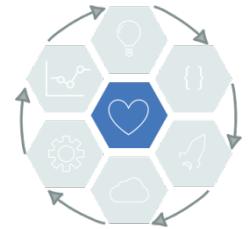




Developers and Operators



개발자에게 DevOps?



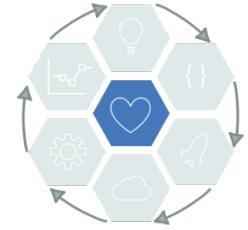
Pros

- Technical dept 감소
- 개발 스킬 증대
- 좋은 개발도구 사용

Cons

- 업무 강도 증가
- Learning curve
- 도구 파편화

그래도 DevOps



Legacy

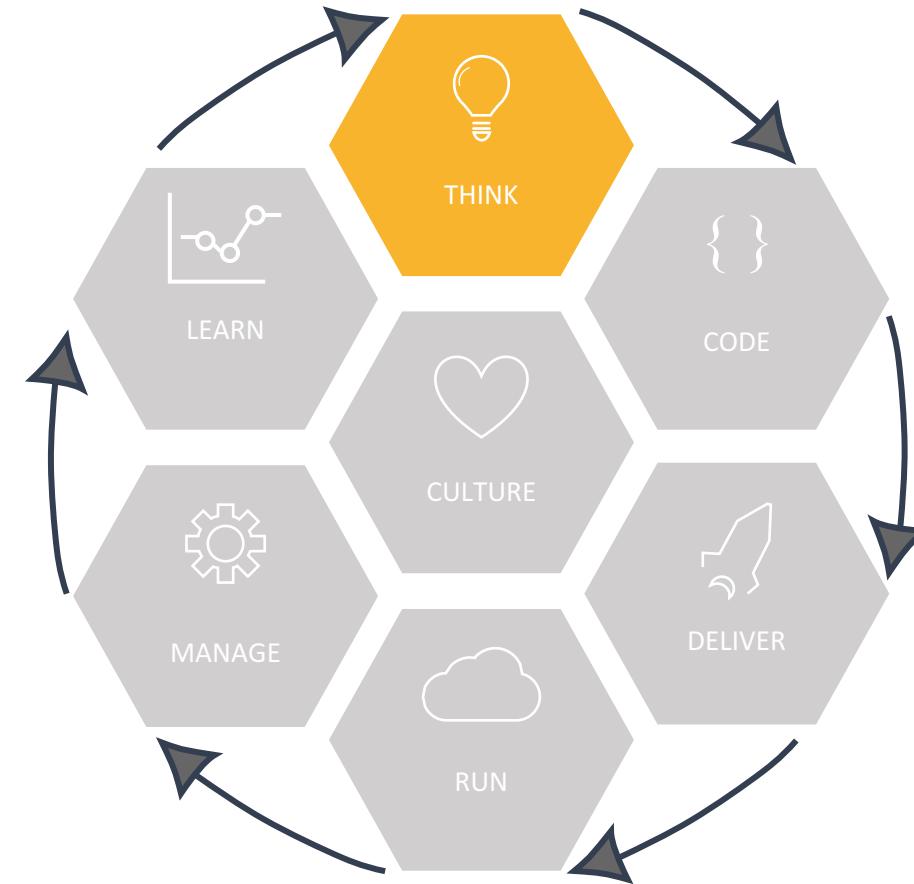
- 대규모 업데이트의 영향 최소화를 위해 새벽에 배포
- 버그 추적과 전체 롤백의 어려움
- 문제점 파악 및 해결에 따른 부서간 반목

DevOps

- 효율적인 테스트 및 디버깅을 위해 업무시간중 배포
- 문제 발생시 바로 롤백, 수정 및 재배포
- 팀원간 원활한 소통과 책임 분배
- 빠른 개발 및 적용

THINK

Incrementally deliver awesome solutions



Issue 전달방법



Legacy

- 구두
- 이메일
- Excel / Word

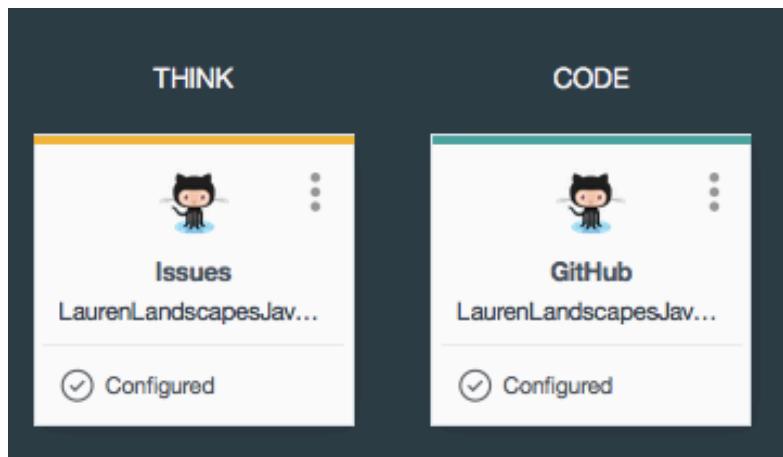
DevOps

- Bugzilla
- Redmine
- JIRA
- GitHub Issue
 - ZenHub
- Trello
- mantis

GitHub Issue



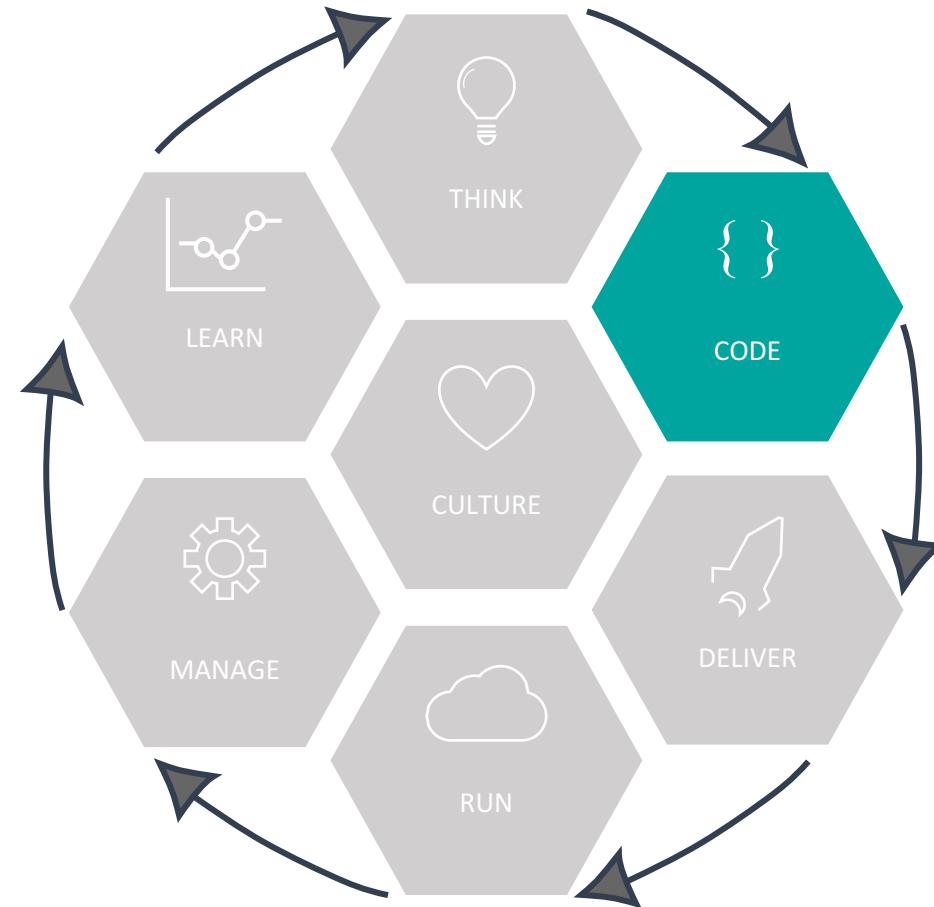
- ▶ GitHub Issue - Issue Tracking 을 위한 도구, GIT과 연동, 손쉽게 이슈 트랙킹 사용



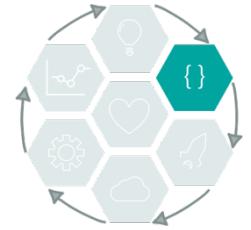
- GitHub Issue에서 이슈 생성 - 담당자 할당 및 카테고리 지정
- 소스코드 Commit 시 'Close, Fix, Resolve' 와 같은 키워드 사용하여 이슈 클로즈
- 이슈-소스코드 연동

CODE

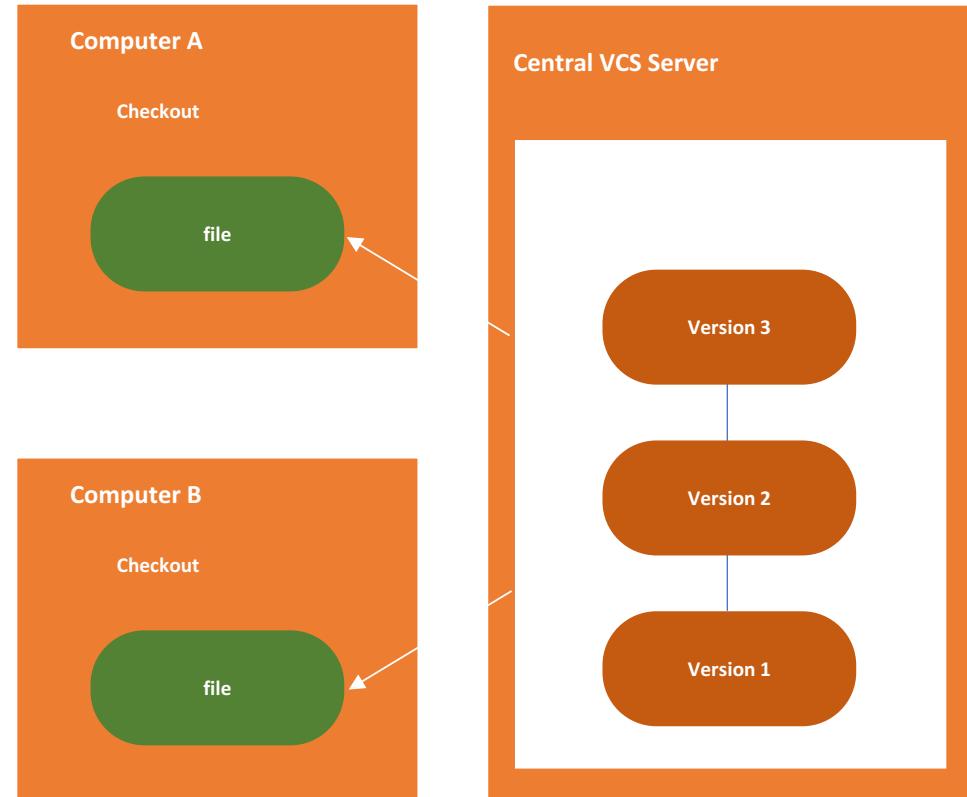
Create innovative solutions fast



Source Code Management (SCM)



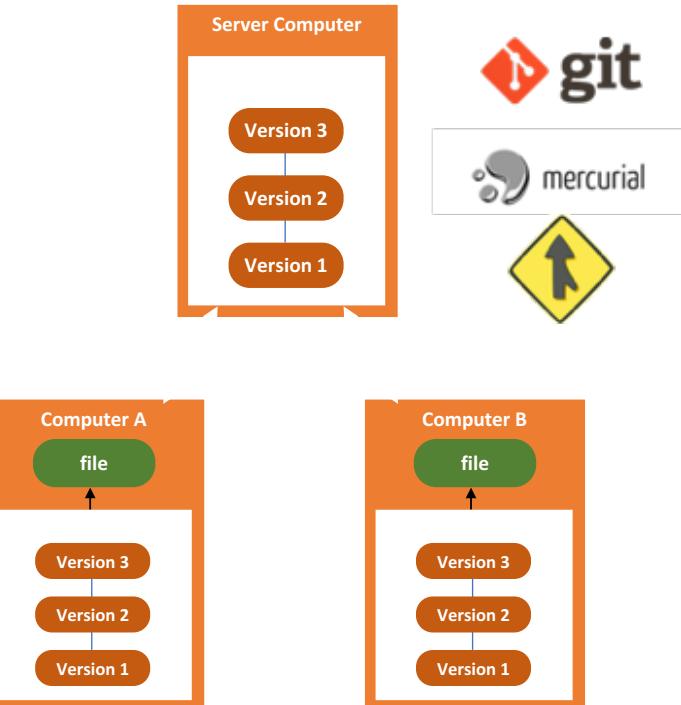
- Version Control System
 - CVS
 - Subversion
- History management
- Branch
- 협업



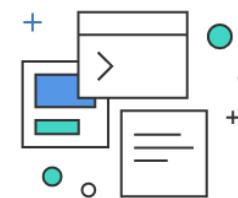
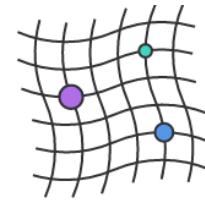
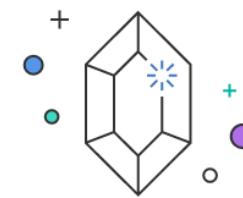
Distributed SCM



- 분산 구조
- git
- mercurial
- bazaar
- 각자 version control 가능
- 쉬운 branch
- 쉬운 commit



IBM Internal Transformation Project



도구는 거들뿐

사례의 적용

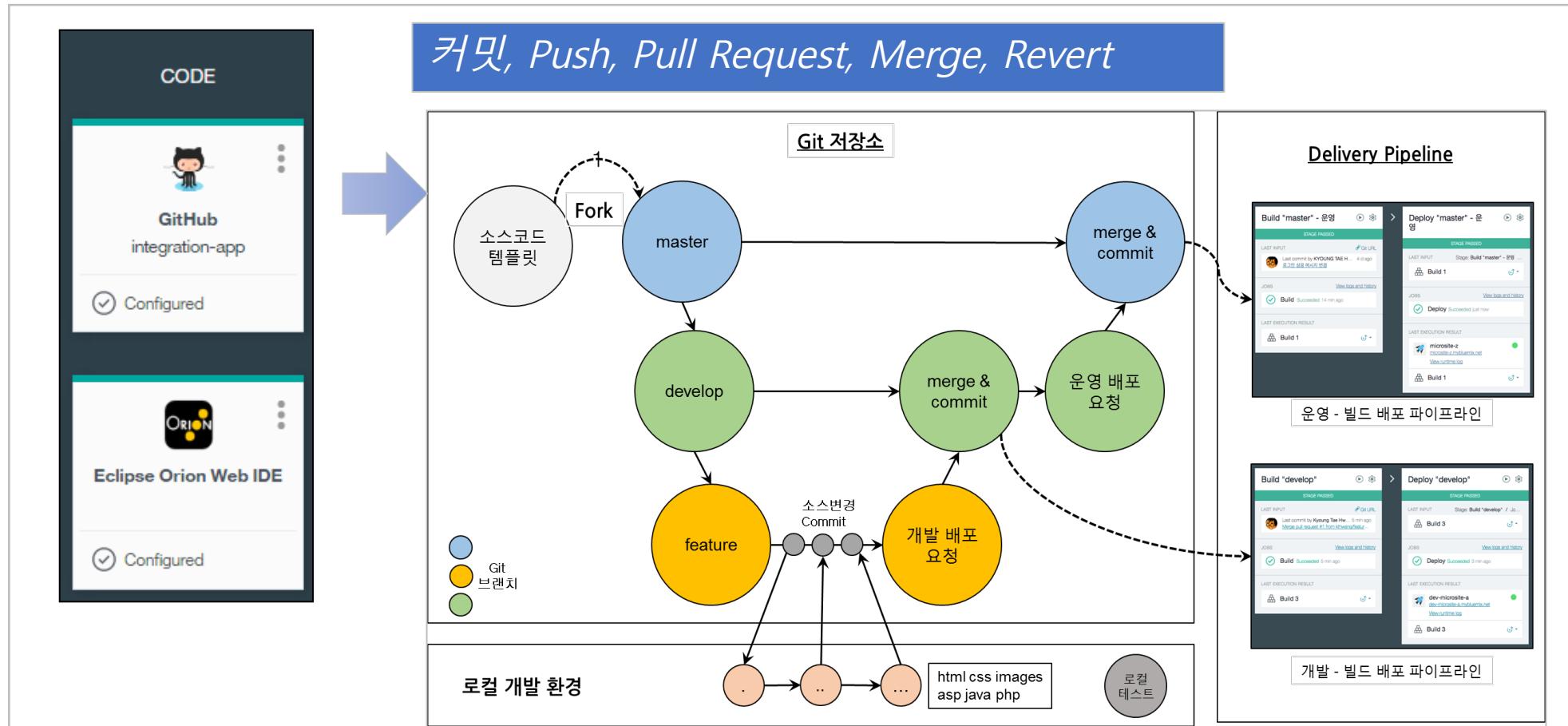
모든 팀의 협업



9개월만에 200+ 팀과 10,000+ 사용자가 GitHub Enterprise 사용

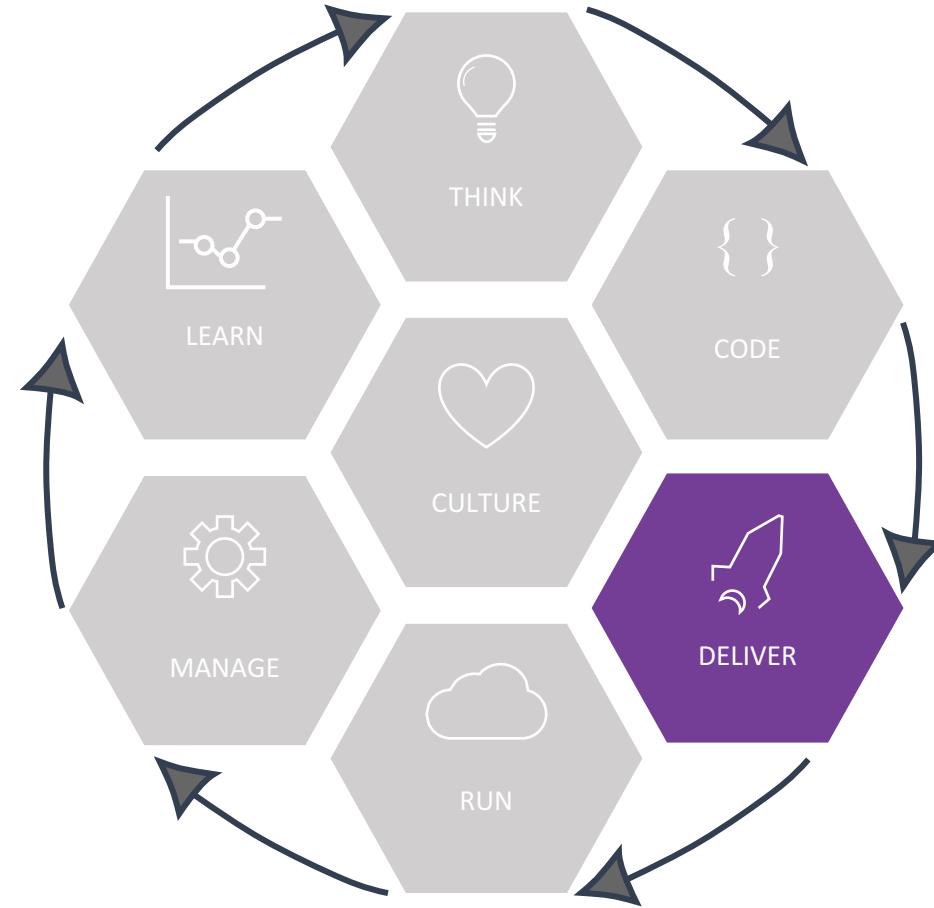
<https://www.ibm.com/devops/method/category/devOpsTransformation/ibm>

Git Branch 관리와 연동

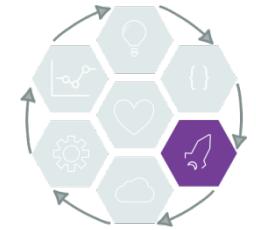


DELIVER

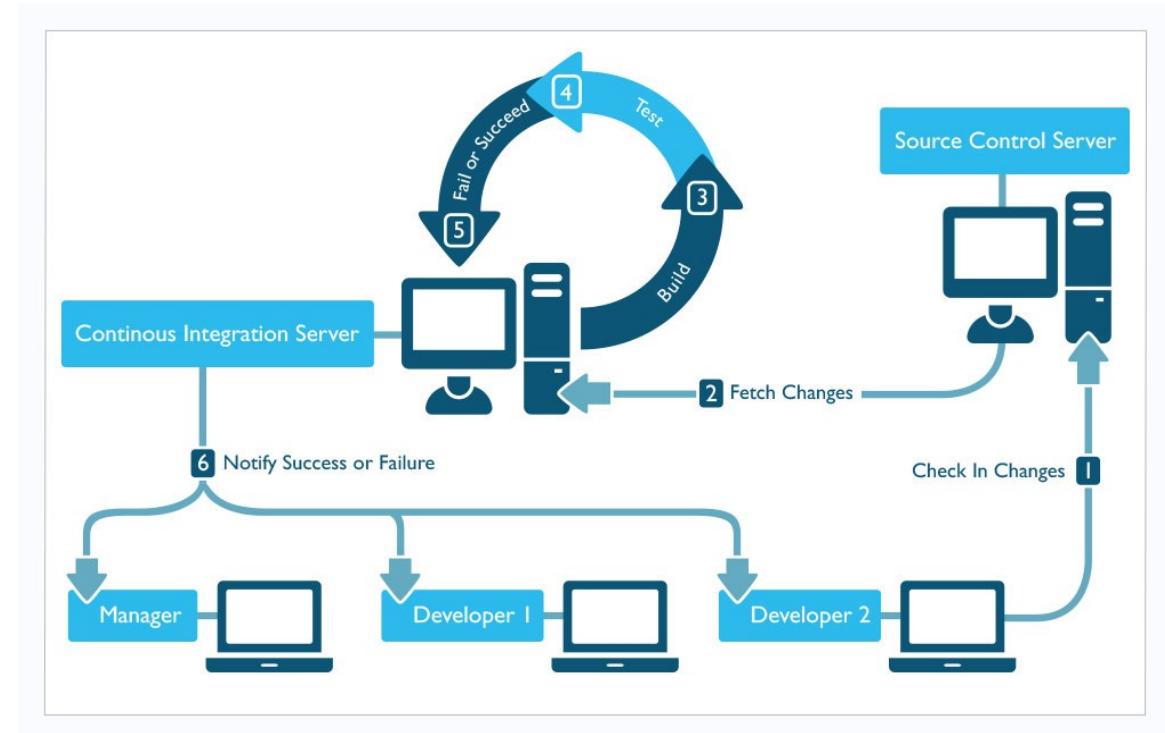
Build, test and deploy



Continuous Integration (CI)

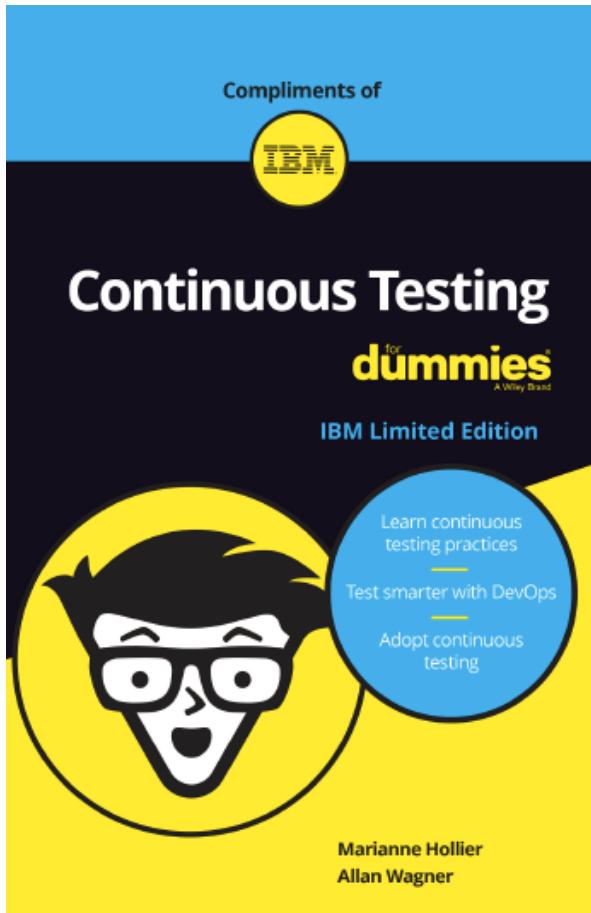
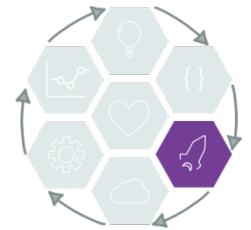


- Test & Push & Test
- Coding style
- Unit test
- Code coverage
- Static analysis
- Functional test
- UX test
 - Selenium
 - PhantomJS



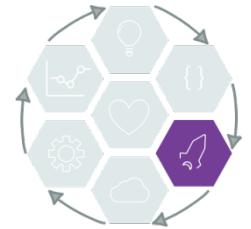
<https://insights.sei.cmu.edu/devops/2015/01/continuous-integration-in-devops-1.html>

Continuous Testing For Dummies - IBM



- [https://www-
01.ibm.com/common/ssi/cgi-
bin/ssialias?htmlfid=KUM12367
USEN](https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=KUM12367USEN)
- 영어, 53 pages

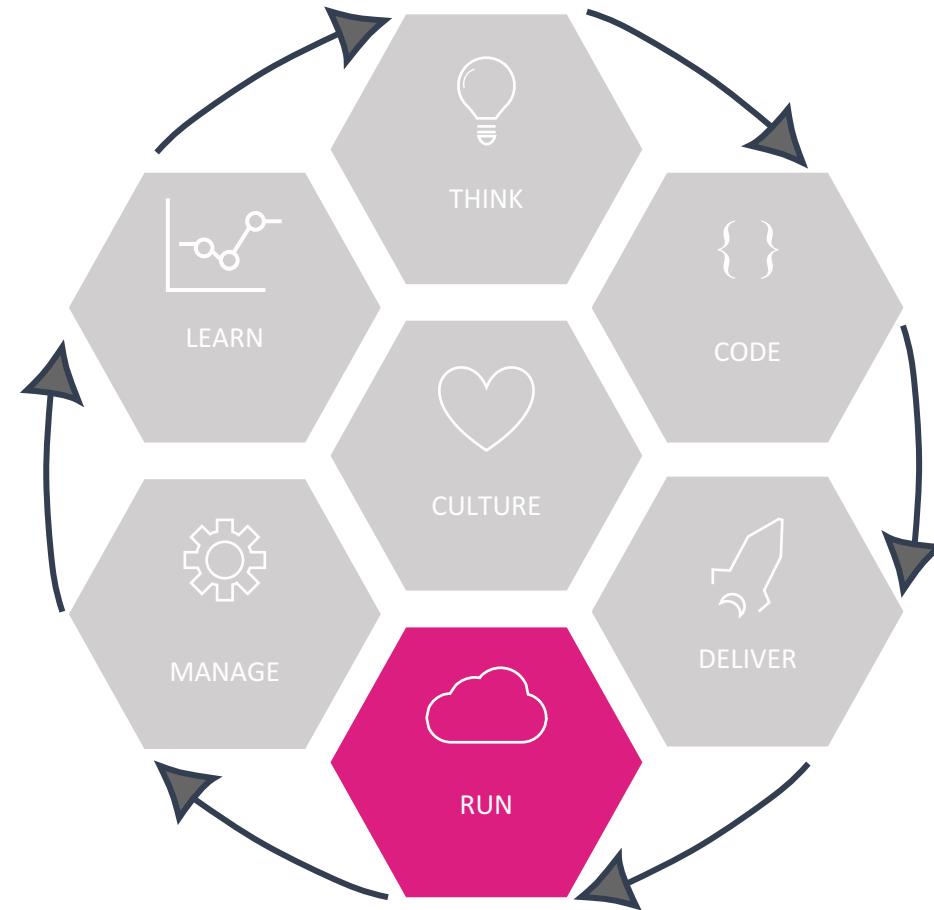
Continuous Delivery (CD)



- CI, CD
 - Jenkins, Travis, UrbanCodeDeploy (IBM)
- Infrastructure 자동화
 - Puppet, Ansible, Chef
- Build / Environment
 - Ant, Maven, Gradle, UrbanCodeRelease (IBM)
 - bower, grunt
- Fabric, npm
- Shell scripts..

RUN

Services, options, and capabilities to run solutions





캠핑의 종류에 따라 준비할 것들?



산, 바다, 계곡



캠핑장



글램핑



호텔

배포 환경에 따라 준비할 것들



Traditional IT	Infrastructure as a Service	Platform as a Service	Software as a Service
Code	Code	Code	Code
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

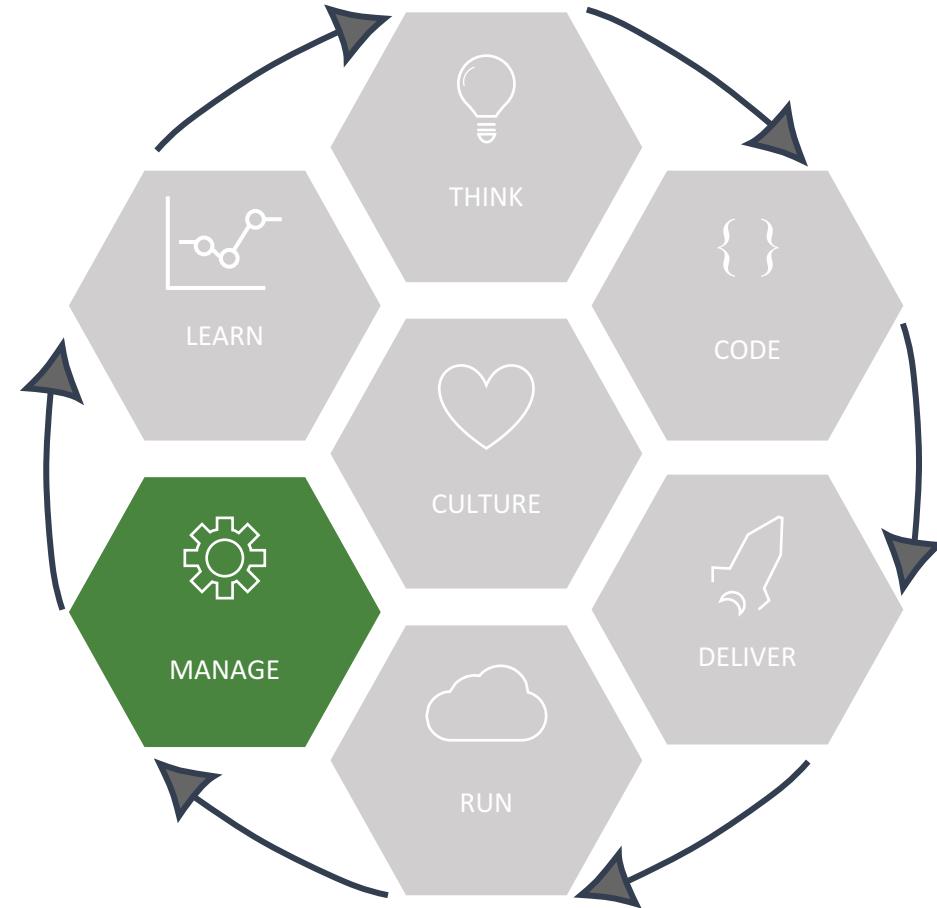
Bluemix 아키텍쳐



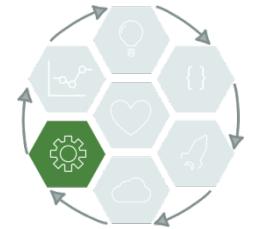
+

MANAGE

Ensure operational excellence



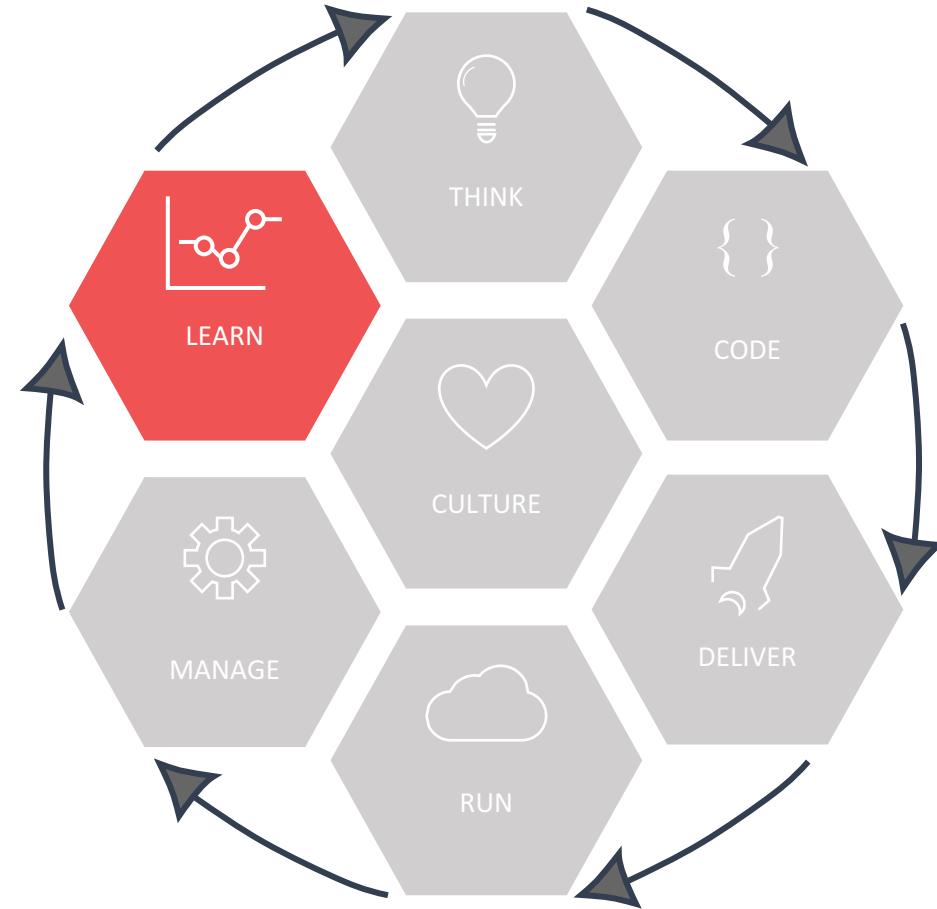
관리 방법론 / 도구



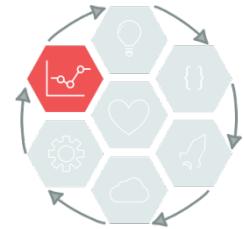
- Fast recovery
- High availability
- Auto scale
- The Circuit Breaker pattern
- Automated monitoring
- Resiliency
- Chaotic testing
- NewRelic
- PagerDuty
- Active Deploy
- Monitoring & Analytics

LEARN

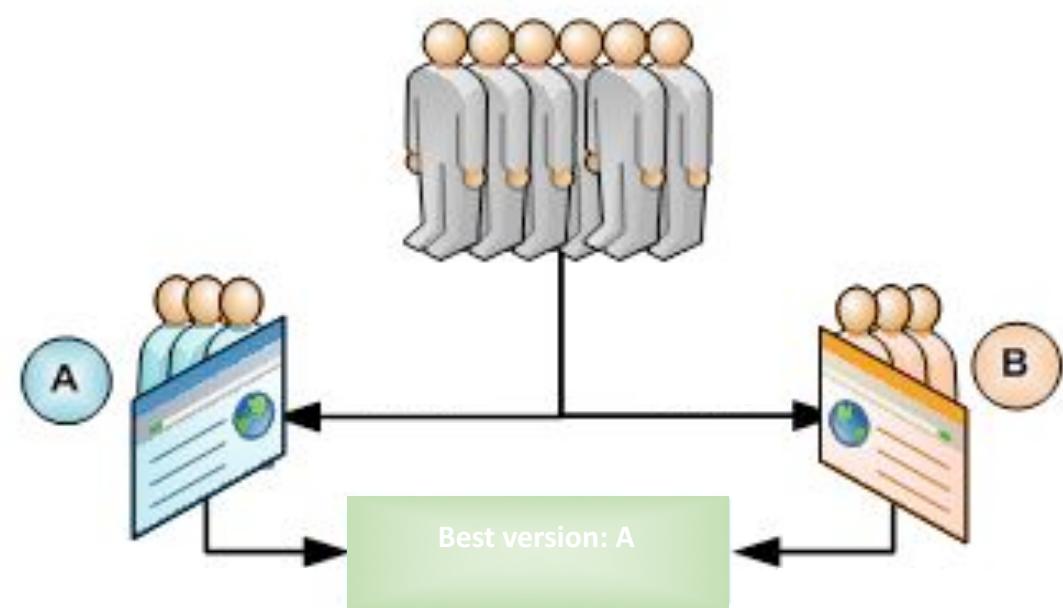
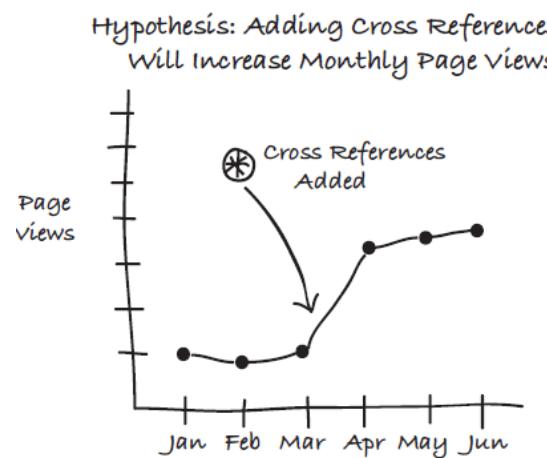
Continuously experiment to deliver the right solution



사용자가 원하는 것?

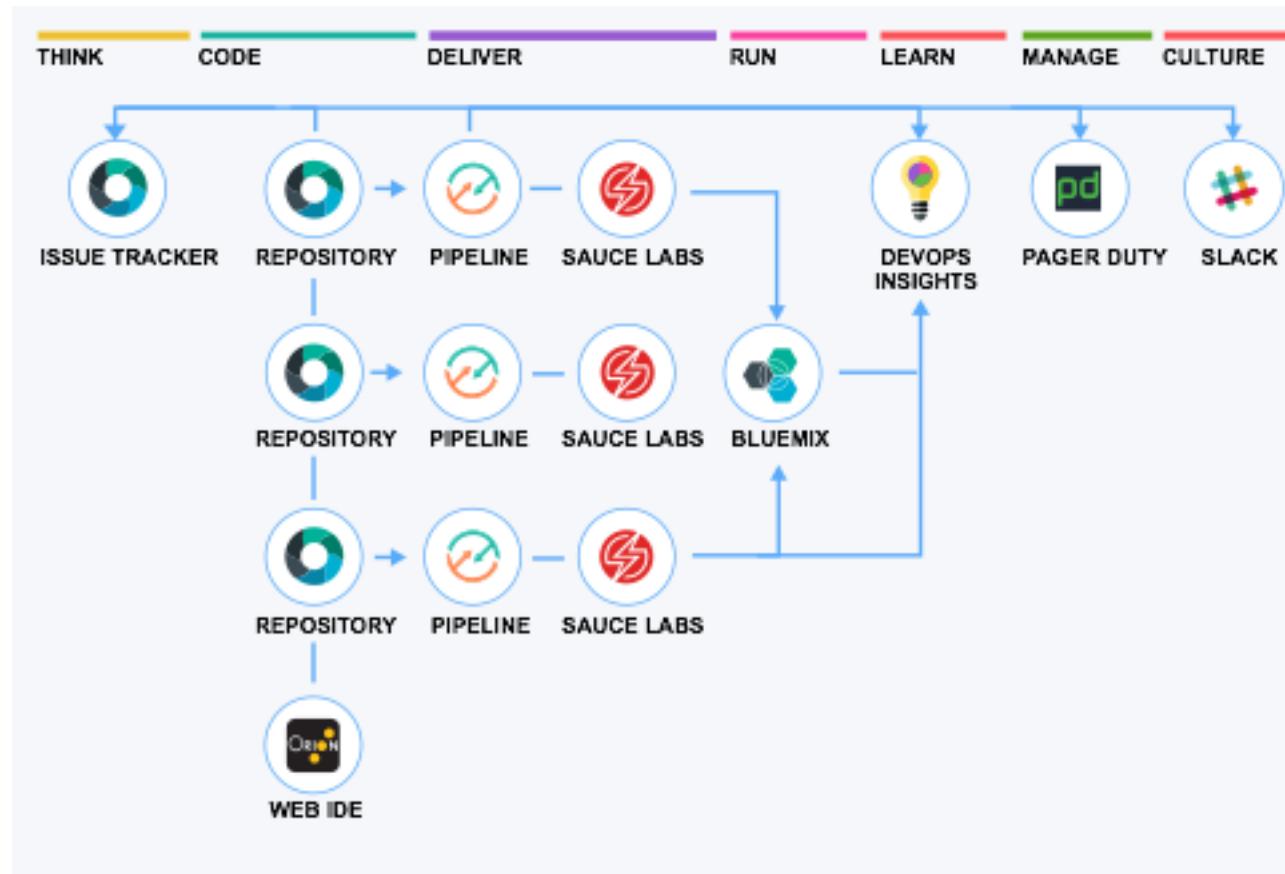
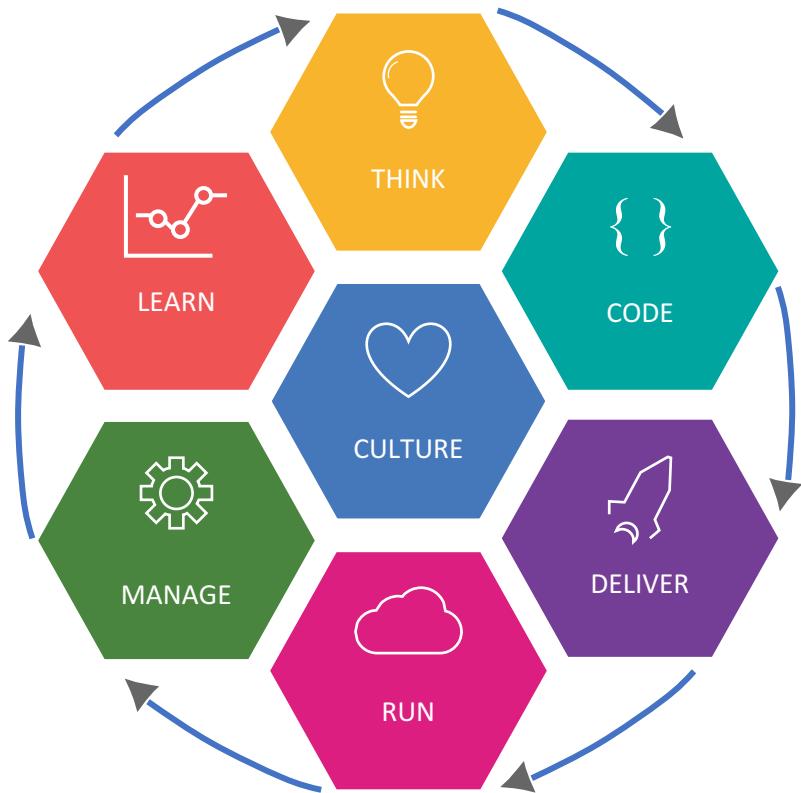


- A/B Testing
- Hypothesis-driven development
- 잘 모르겠으면 다 구현해보고 사용자의 반응을 보자



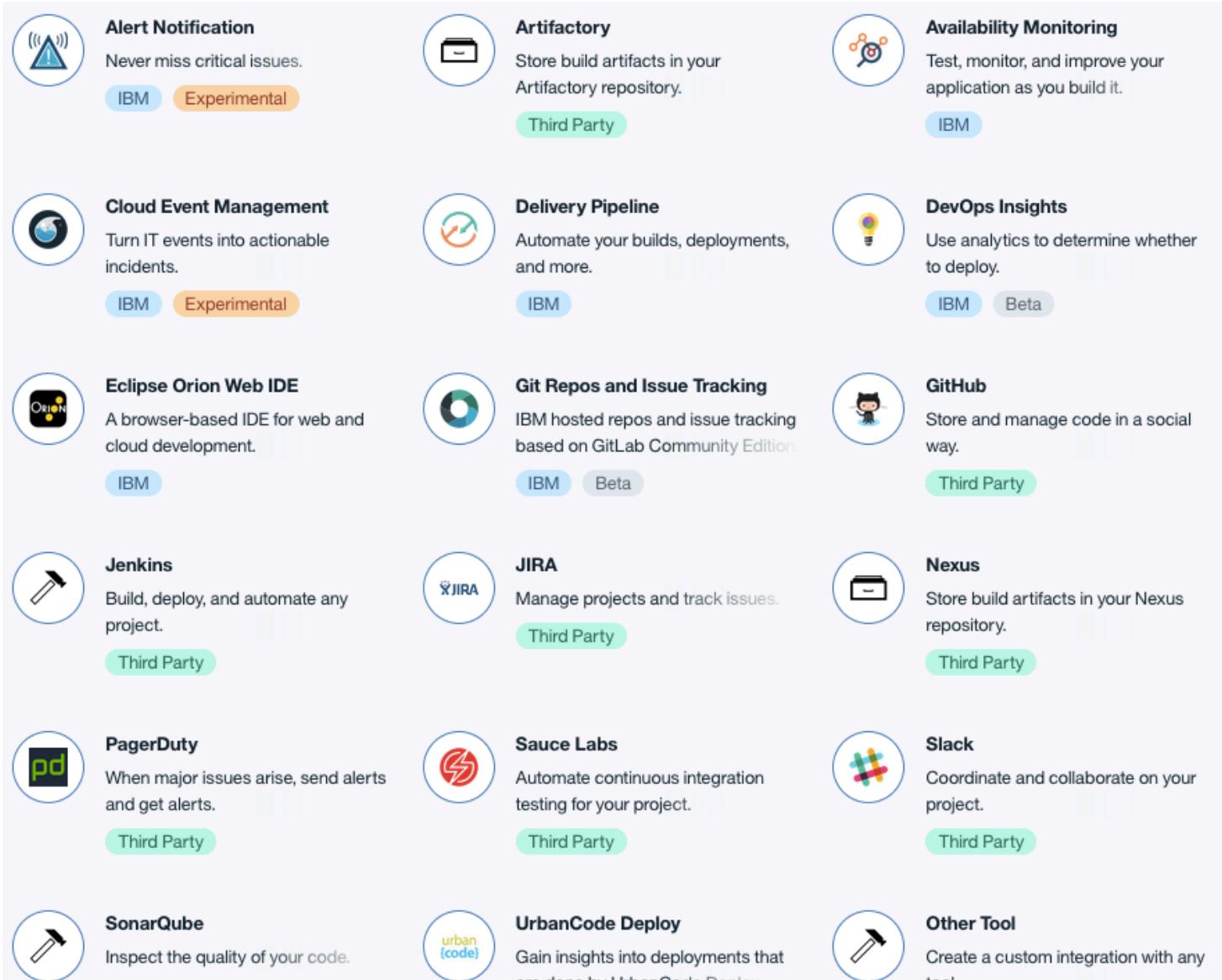
<https://www.ibm.com/developerworks/library/d-testops-continuous-test/>

Open Toolchain



Toolchain

- DevOps에 사용되는 도구를 모아 통합
- $1 + 1 + 1 > 3$
- 프로젝트, 개발, 제품의 특성에 따라 툴체인들은 서로 다름



Open Toolchain

- Toolchain template 을 통한 best practice 적용 가능
- Custom template 을 제작하여 다른 프로젝트에 toolchain 적용
- Toolchain API 및 SDK 를 통해 다른 도구 연동 가능 (공개 예정)

Continuous Delivery Templates

 Garage Method Cloud-native Tutorial toolchain Apply practices and tools across the DevOps lifecycle.	 Microservices toolchain with DevOps Insights Continuously deliver a microservices app.	 Microservices toolchain with DevOps Insights (v2) Continuously deliver a microservices app with repos and issue tracking
 Secure container toolchain Continuously deliver a secure Docker app.	 Simple Cloud Foundry toolchain Continuously deliver a Cloud Foundry app.	 Simple Cloud Foundry toolchain (v2) Continuously deliver a Cloud Foundry app with repos and issue
 Simple Cloud Foundry toolchain with DevOps Insights Use analytics to determine whether to deploy.	 Simple container toolchain Continuously deliver a Docker app.	

DevOps Insights Templates

 Delivery Insights with IBM UrbanCode Deploy View deployment metrics.	 Deployment Risk Analytics with GitHub and Jenkins Use analytics to determine whether to deploy.	 Developer Insights and Team Dynamics with GitHub and JIRA Analyze GitHub source code, and GitHub or JIRA issues.
---	--	---

Other Templates

 Build your own toolchain For advanced users, create your toolchain from scratch.

IBM Bluemix Continuous Delivery

The screenshot shows the IBM Bluemix Continuous Delivery interface. It features two main sections: 'Start with a pipeline' and 'Start from a toolchain template'. The 'Start with a pipeline' section includes a 'Start here' button and a note about deploying apps with automated build and delivery. The 'Start from a toolchain template' section includes a 'Start here' button and a note about creating a toolchain for planning, developing, and managing apps. Both sections show a preview of the Bluemix interface with various tools like Build, Deploy, and Monitor.

Get started with Continuous Delivery

Start with a pipeline

Start from a toolchain template

Already have toolchains? [View your toolchains.](#)



통합된 DevOps toolchain
작성



막힘없이 배포하고 통합
파이프라인 관리



DevOps Insight 를 통한
품질 관리



Web IDE 를 통한 수정



Git 소스 관리 및 이슈
추적

Demo

Questions?

Thank you!

References

- IBM Cloud Garage Method <https://www.ibm.com/devops/method/>
- IBM 한국 기술 포럼 <https://developer.ibm.com/kr/>
- Bluemix 사용자 그룹
<https://www.facebook.com/groups/BluemixStudy/>
- Open Toolchain 시작하기
<https://console.ng.bluemix.net/devops/create>
- Open Toolchain repository <https://github.com/open-toolchain>