

○ 과정에 들어오면서 문득 든 궁금증

- 우리는 예전부터 Jenkins로 빌드/배포를 하고 있었는데 뭐가 달라졌길래 교육까지 들어야하지?
- CI/CD가 그렇게 중요해? 왜 여기저기서 CI/CD란 단어가 보이지?
- Jenkins + Maven, Ant 조합이면 빌드하고 배포하고 모든 걸 다 할 수 있었는데?
- 자동화? 그럼 어디까지 자동화 할 수 있지?

Module 1. CI/CD 개요

1.1 CI/CD 정의

1.2 환경의 변화

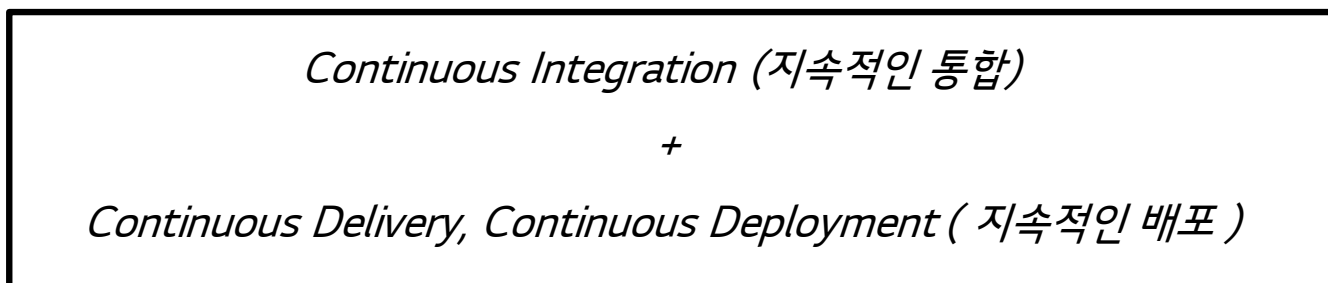
1.3 문화의 변화

1.4 CI/CD 패러다임 변화

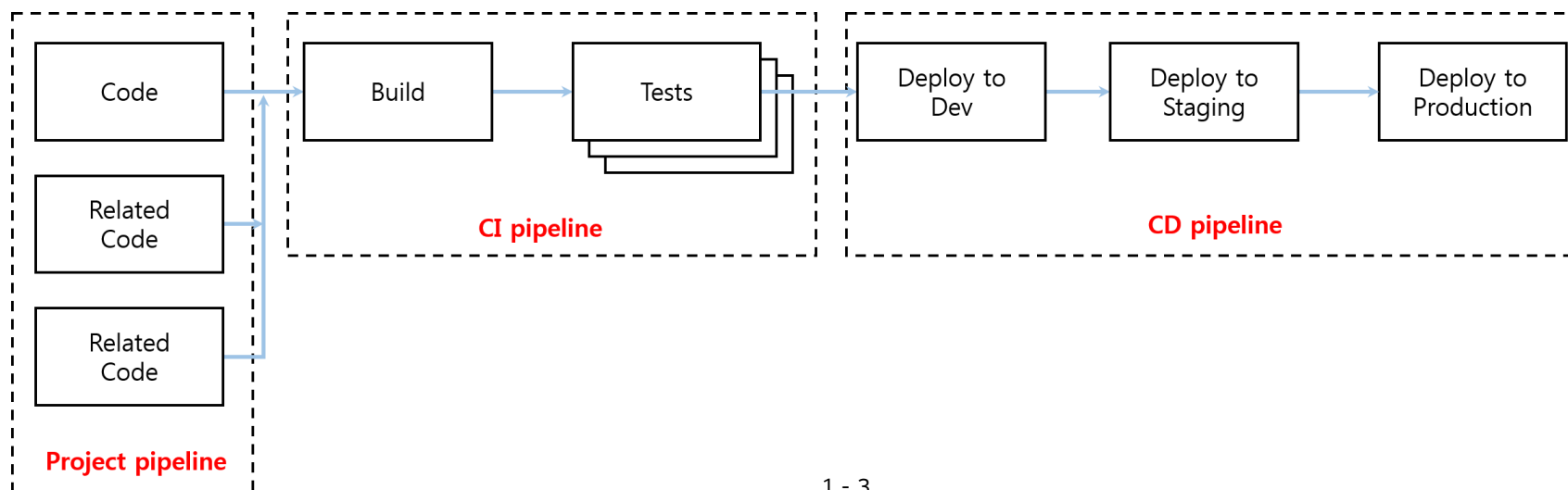
1.5 CI/CD 목표

Summary

In software engineering, CI/CD or CICD generally refers to the combined practices of continuous integration and either continuous delivery or continuous deployment - Wikipedia



어플리케이션 빌드 및 배포와 인프라 Provisioning을 자동화하여 산출물을 짧은 주기로 고객에게 제공하는 방법





Jenkins



MavenTM

 **Nexus**





환경의 변화

Business Agility

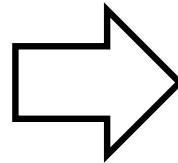
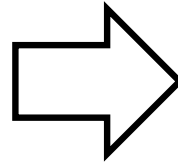


- Business Agility(민첩성): 빠른 개발, 적용, 검증, 확대 > 빠르고 다양한 사업환경 변화 및 고객의 요구사항에 적시 대응하여 비즈니스 경쟁력 확보
- Rapid/Frequent/Reliable Software Delivery, Application Architecture 방법 중 하나로 MSA
- 위의 목적이 아니거나, 현재 Monolith 구조에서도 문제가 없다면 MSA 도입 재고 필요

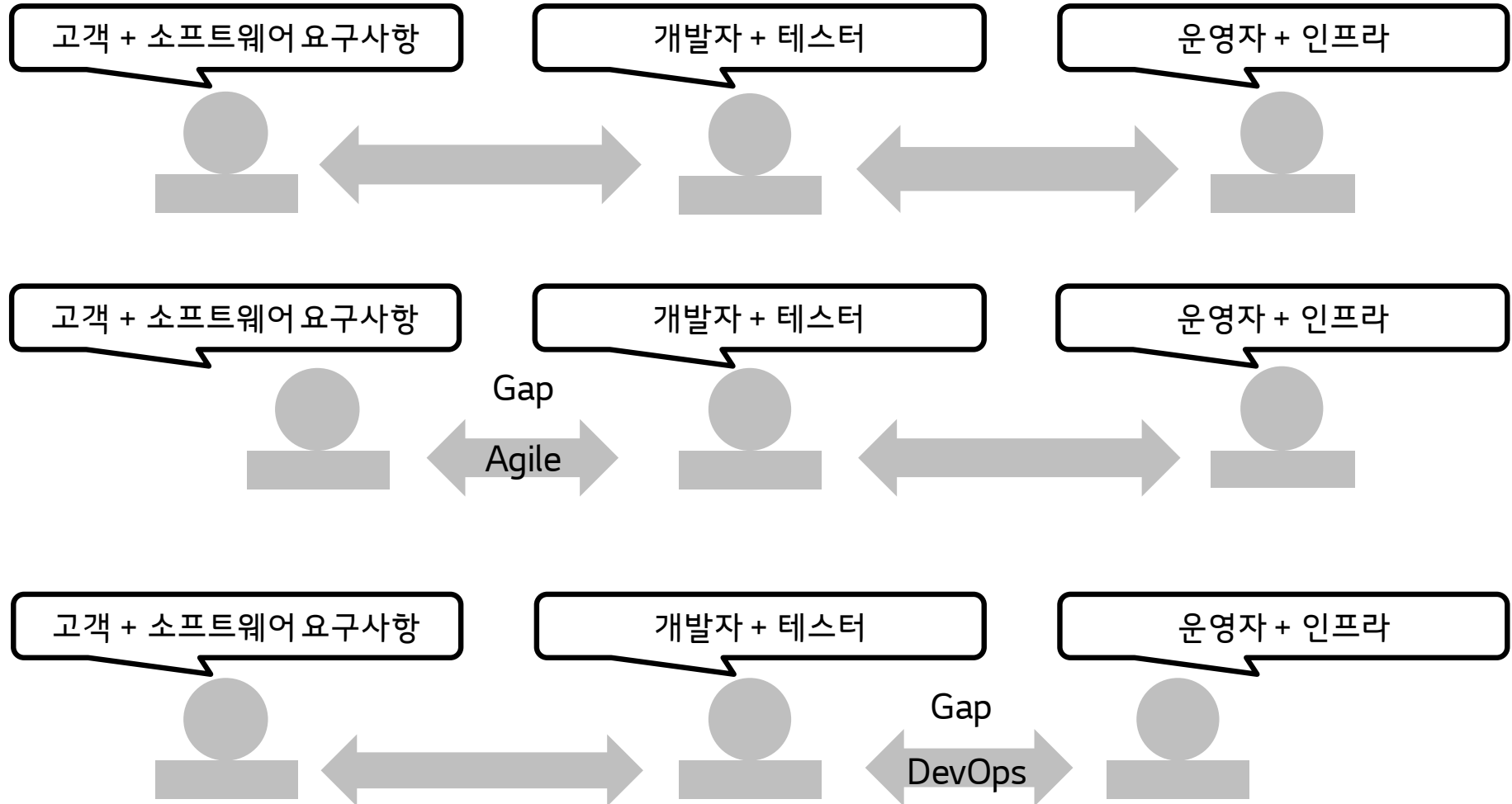
- 필요하면 바로 사용 (자원 효율성 증가)
 - 서비스 생성/폐기 용이
 - 사용한 만큼만 비용 지불
- Auto Scaling
- 보안, API Gateway, Monitoring 등 부가 기능 활용 용이
- Main Frame > Client/Server > Web 으로 발전과 같은 Cloud로의 Paradigm Shift 필요

- DevOps: 개발 따로, 운영 따로 > 운영하면서 지속적으로 개발, 개발(Development)과 운영(Operation)을 결합한 합성어로 개발, 빌드, 테스트 및 운영이 용이하도록 사고방식의 변화, 협업 향상 및 긴밀한 통합을 강조
- 기업의 조직과 문화 등 기술 외적인 변화도 수반되어야 함

Cloud 기반 및 DevOps 조직이 뒷받침 되어야
MSA 도입 목적 및 Business Agility 극대화

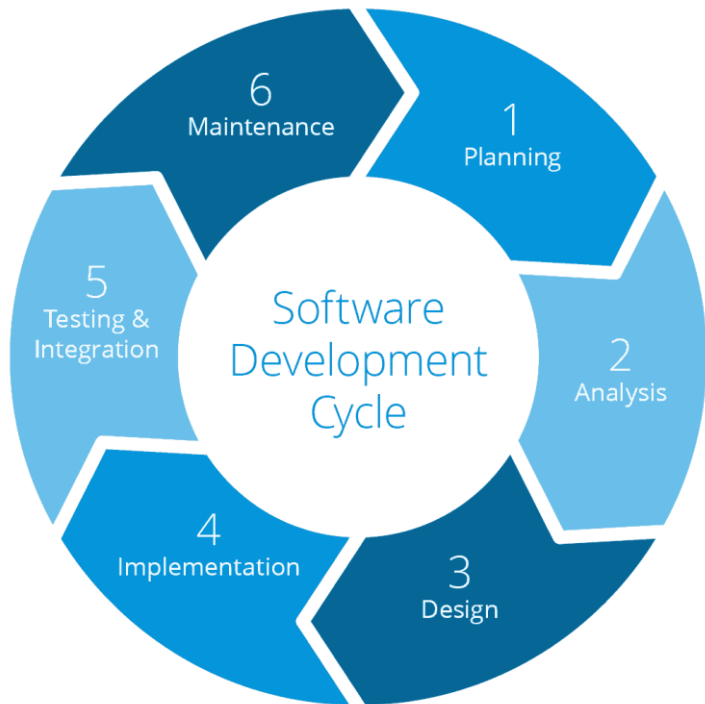


문화의 변화

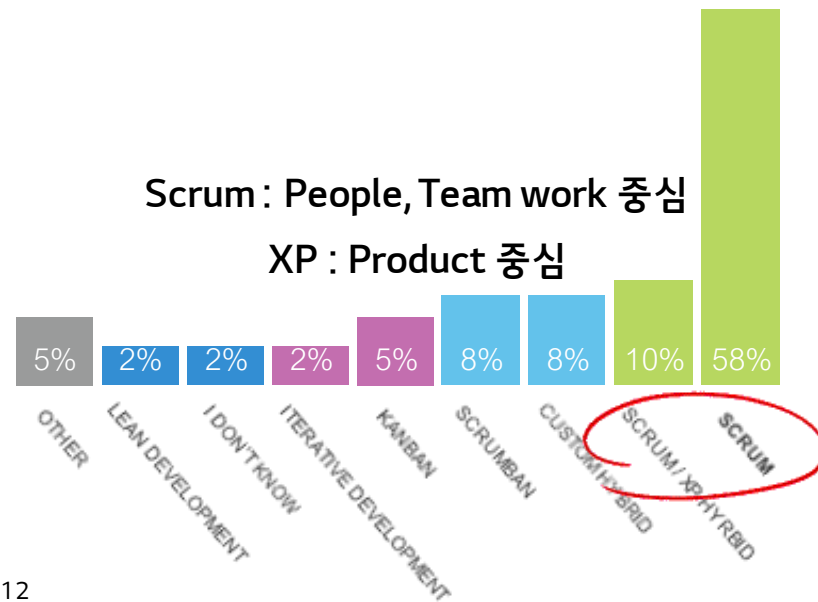


Agile 이란?

고객 가치 극대화를 위해 최선의 의사결정을 내리는데 도움이 되는 가치와 원칙이며 소프트웨어의 불확실성이 증가됨에 따라 개발 방식을 프로토타입과 학습에 초점을 맞춘 방법론



Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan



DevOps 란?

소프트웨어 개발자들과 IT종사자들 사이의 의사 소통, 협업, 융합을 강조한 소프트웨어 개발 방법론



DevOps의 조직체계/문화, 프로세스, 기술 : CAMLS

Culture	Automation	Lean	Measurement	Share
<ul style="list-style-type: none"> ✓ 실험을 통한 학습과 협업을 중시 ✓ 'Do Not Fail'이 아니라 'Fail Early' 	<ul style="list-style-type: none"> ✓ 'Continuous Delivery' 단계를 최대한 자동화-> Test 자동화/IAC ✓ 단계별 자동화에서 시스템 개발~운영 전체관점의 자동화 	<ul style="list-style-type: none"> ✓ 작은 배치 사이즈(Small Batch Sizes), 사용자 Value 창출 중심 ✓ Big Batches 에서 → Small Batch 로 	<ul style="list-style-type: none"> ✓ 시스템 지표, 비즈니스 지표(A/B Testing) 등 모든 것을 모니터링 ✓ 시스템 가용성 중심 지표에서 MTTR¹⁾, 사용자 접속자 수 	<ul style="list-style-type: none"> ✓ 공통 목표, 협업을 위한 원활한 의사소통 CSR → ChatOps (Slack 등) ✓ 비즈니스 지표(A/B Testing) 등 모든 것을 모니터링 ✓ 기능중심의 Functional 조직에서 서비스 중심의 Cell 조직

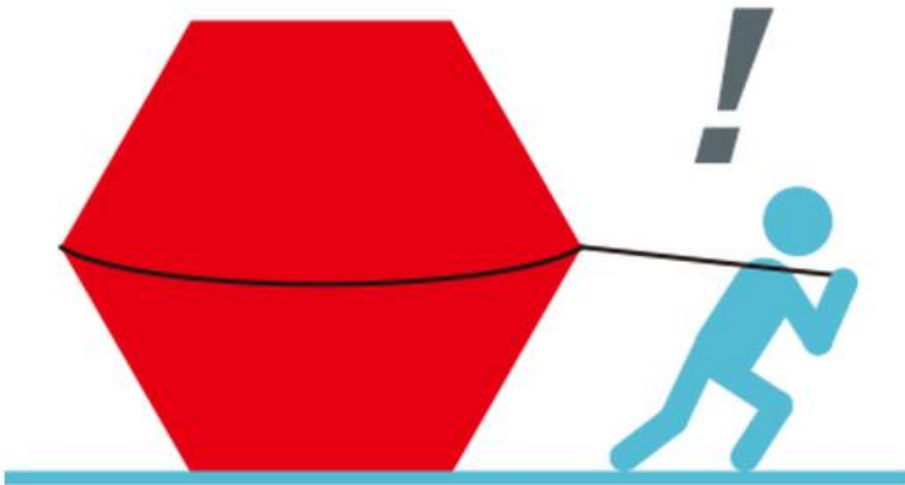
Agile	CI/CD	DevOps
프로세스 문화에 초점	소프트웨어 라이프 사이클에 초점	개발 문화에 초점
변화를 강조	도구를 강조	역할을 강조



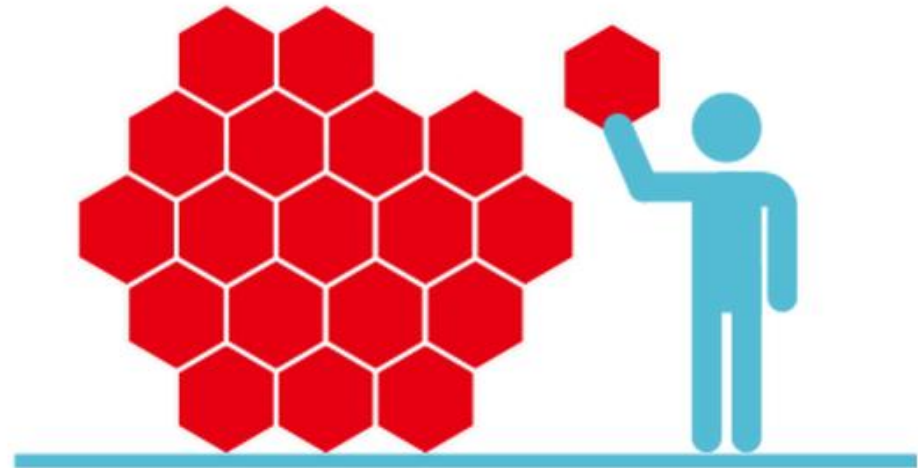
다른 영역이라고 생각해야 할까요?

<https://youtu.be/5Eqz8hm3SVQ>



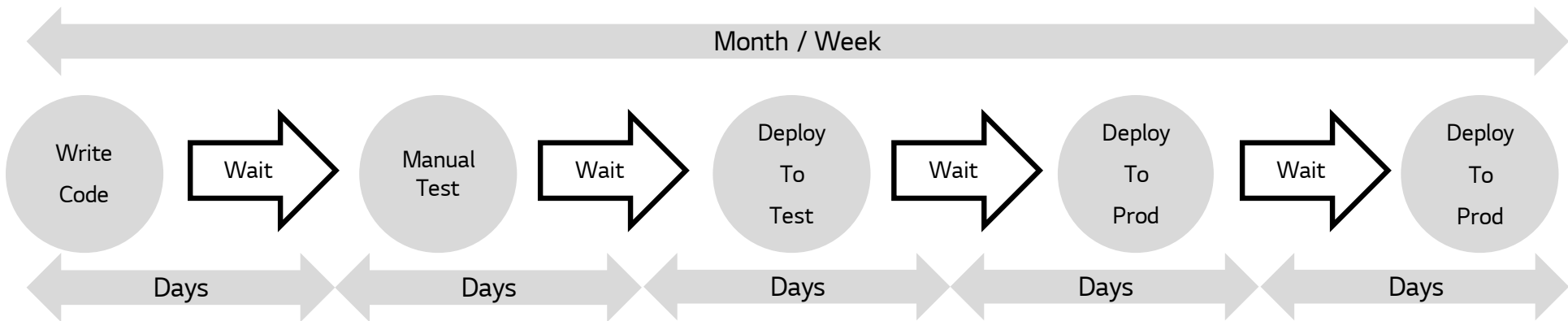


*This project has got so big,
I'm not sure I'll be able to deliver it!*

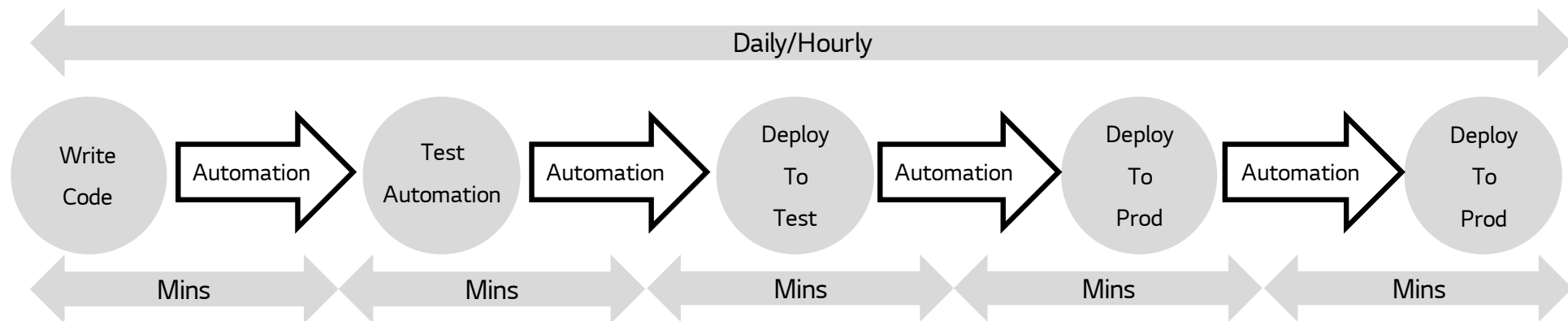


*It's so much better delivering this
project in bite-sized sections*

As-Is : 빌드부터 배포까지 평균 2주 혹은 한 달에 한번 수행



To-Be : 빌드부터 배포까지 일 단위 혹은 시간 단위로 수행



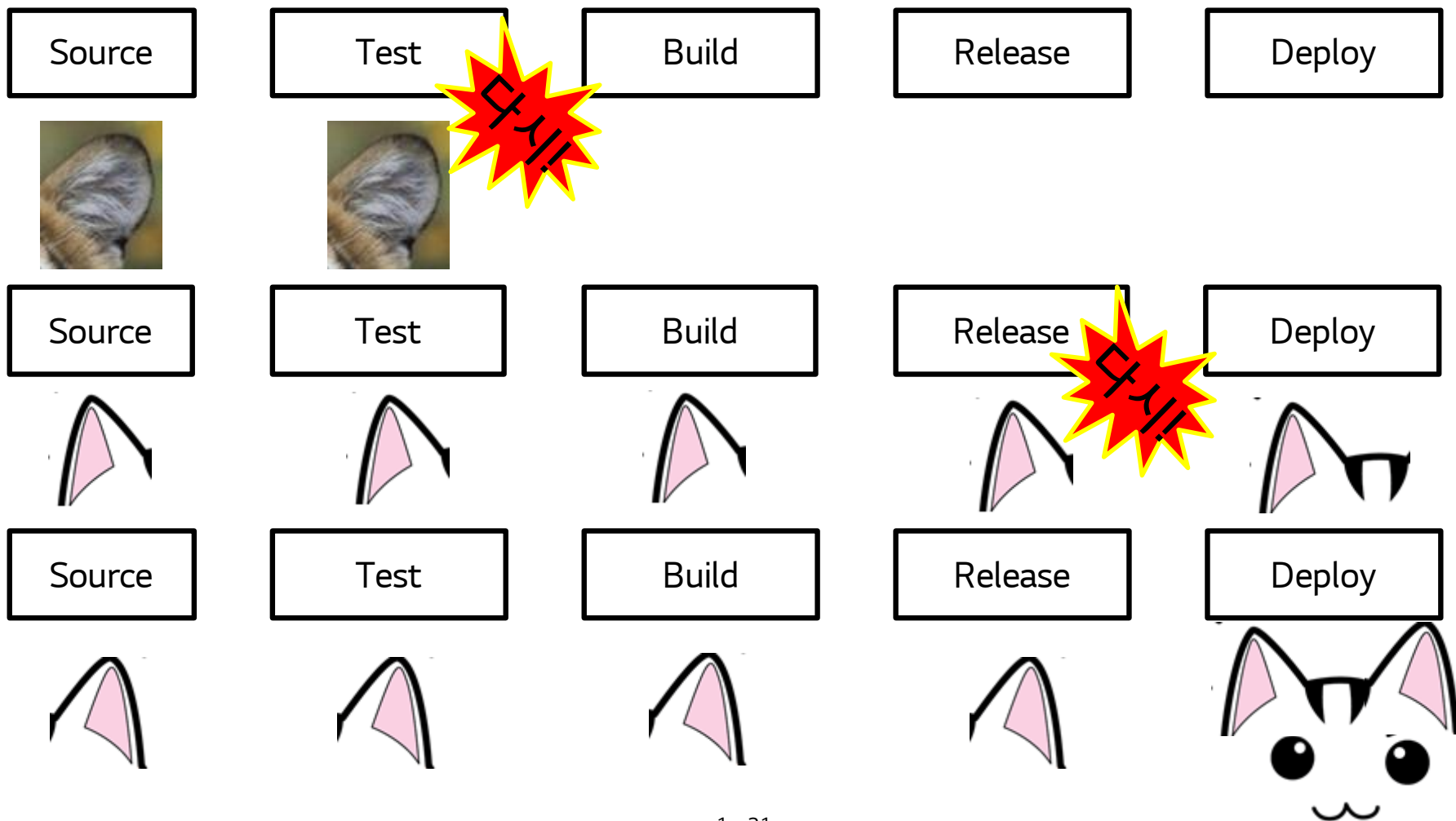


고객이 생각한 캐릭터



내가 생각한 캐릭터

CI/CD는 파이프라인을 구축하여 어플리케이션 개발 단계를 자동화하여 배포 프로세스를 여러 단계로 나누고, 각 단계마다 어플리케이션에 추가 혹은 변경된 기능의 유효성을 검사하고 품질을 다각도로 확인하여 짧은 주기로 지속적으로 제품(릴리즈)을 출시하는 것



CI/CD는 파이프라인을 구축하여 어플리케이션 개발 단계를 자동화하여 배포 프로세스를 여러 단계로 나누고, 각 단계마다 어플리케이션에 추가 혹은 변경된 기능의 유효성을 검사하고 품질을 다각도로 확인하여 짧은 주기로 지속적으로 제품(릴리즈)을 출시하는 것





<https://www.youtube.com/watch?v=53SBQSQTIRU>

IT환경과 개발 방법론의 변화로 인하여 Business Agility를 확보하기 위해서는 Monolith 환경에 비해 복잡한 아키텍처를 구성하여야 하므로 자동화가 고려된 아키텍처 설계 필수

CI/CD는 DevOps와 Agile을 구현하기 위한 자동화의 핵심요소이며 프로젝트 환경의 Outer Architecture 구성에 대한 고민과 이를 자동화 하기 위한 CI/CD에 대한 고민이 필요함