



한국정보과학회
소프트웨어공학 소사이어티
Software Engineering Society

산업 AI를 위한 소프트웨어 공학

KCSE 2021 튜토리얼
2021.2.1

박찬진 (오브젠 AI Lab)



개요

- 소프트웨어 공학 기술은 기업의 디지털 혁신을 위해 중요하게 활용되고 있다. 하지만, 지능형 기업의 단계로 발전하기 위해서는 AI 개발을 위한 엔지니어링 기반을 갖추어야 한다
- 본 튜토리얼에서는 다음 주제를 다룬다
 - Digital Transformation을 위한 소프트웨어 공학
 - AI 개발과 전통적 SW 개발 간의 차이, AI 개발 방법론의 필요성 설명
 - 하이닉스에서의 불량 검사 자동화 과제에 AI 적용을 통해 배운 교훈 소개. AI 운영시스템화가 중요
 - 불량 검사 AI 모델 하나 만으로 완성되지 않으며 다양한 형태의 AI를 필요



Certificate of Appreciation ICPC 2020

In grateful recognition of sharing the knowledge and expertise as a keynote speaker during the 28th International Conference on Program Comprehension entitled

**“Software Engineering for
Industrial AI: A Key Enabler of
Digital Transformation”**

Presented to
Chanjin Park

Eunjong Choi
General Chair

Yann-Gaël Guéhéneuc & Shihpei Hayashi
Program Chairs

Digital Transformation and AI

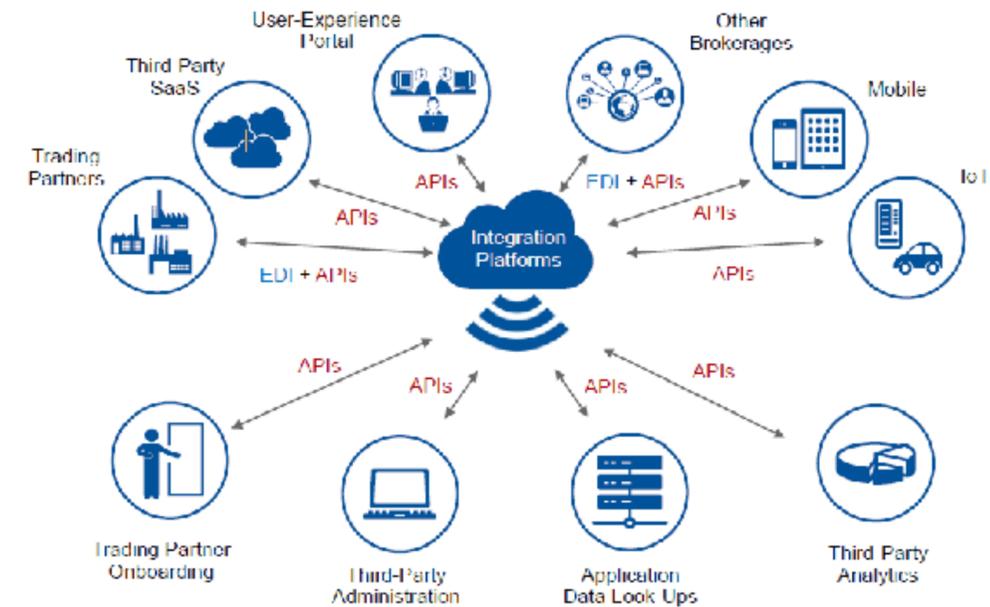
SE for Digital Transformation, 2017

- 소프트웨어 공학 기술을 기존 산업의 디지털화 및 디지털 비즈니스로의 전환을 위해 활용

모듈화와 마이크로서비스화, API Design

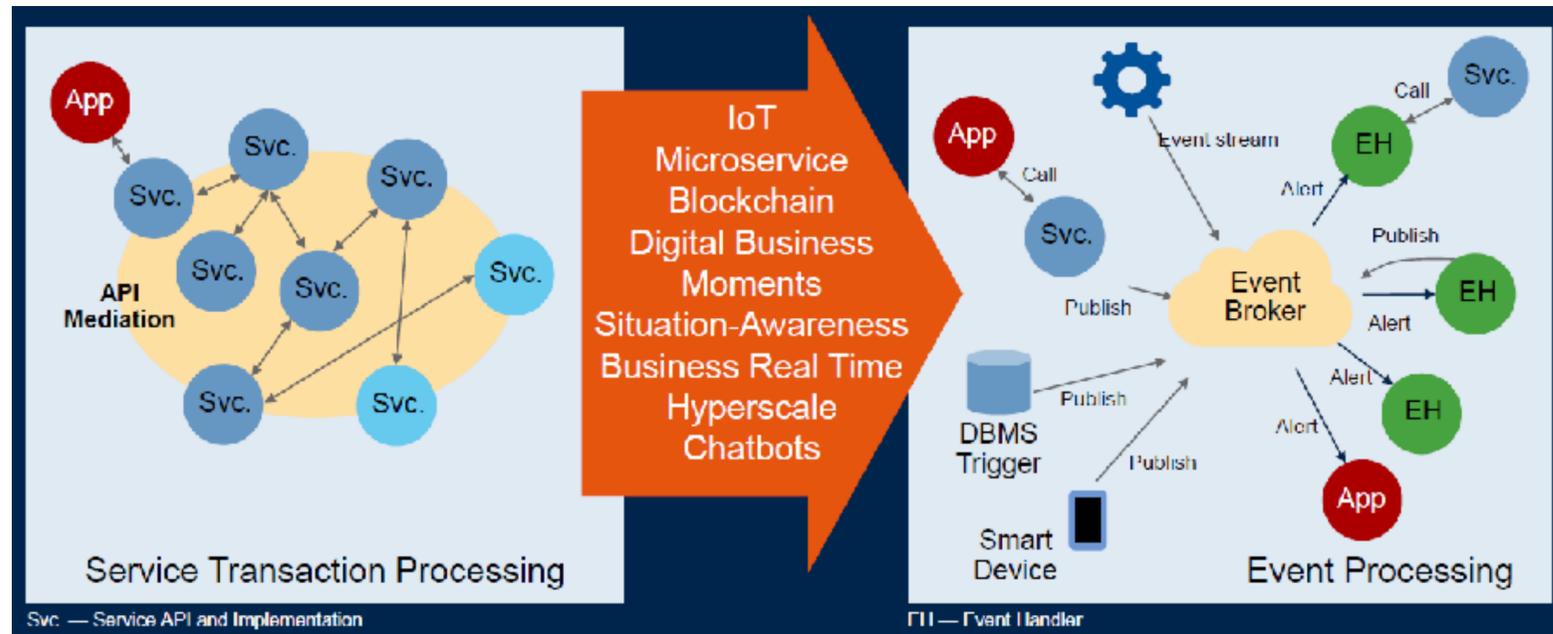


Integration & Service Mesh



이벤트 중심 체계로 전환

Event-Driven (Event Broker, Event Handler, Subscribe & Publish)



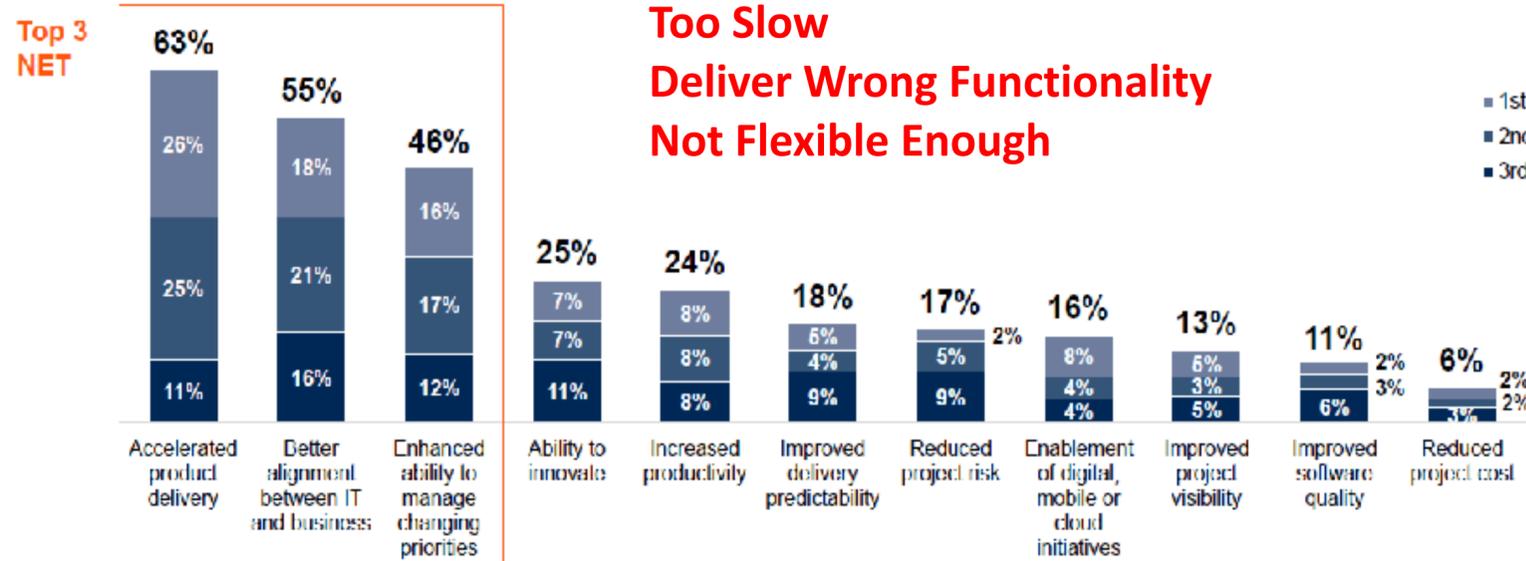
SE for Digital Transformation, 2018

- 변화의 지속성을 강조: ContinuousNext

Agile & DevOps 방법론의 도입이 필수적

The Pressures of Continuous Next Drive Agile and DevOps Adoption

What were or are your organization's top 3 most important reasons for adopting agile development?

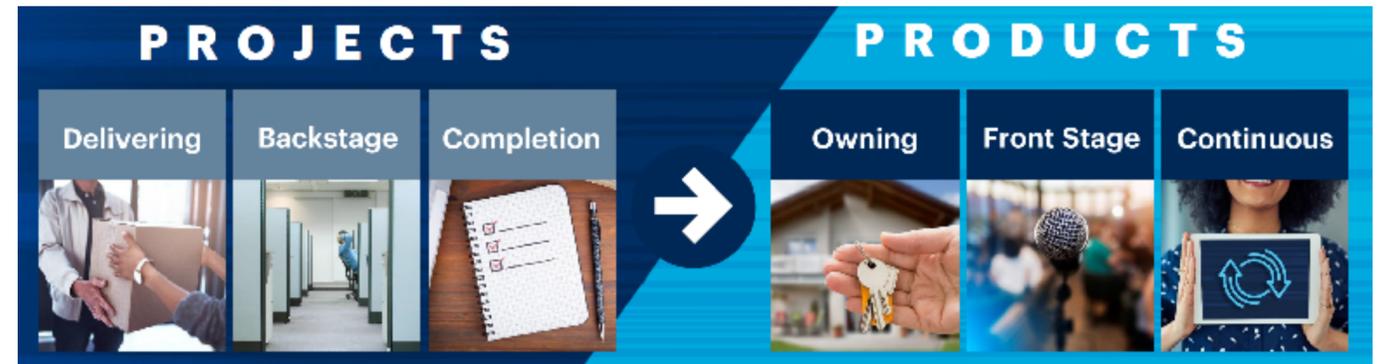


Too Slow
Deliver Wrong Functionality
Not Flexible Enough

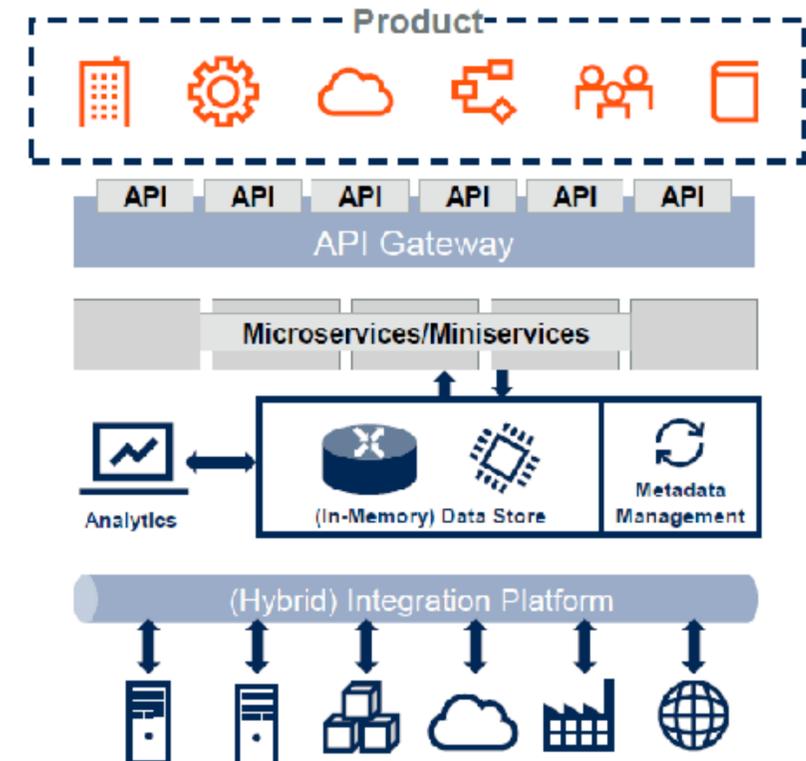
Root Cause is "Deliveries are TOO BIG"

Gartner

Projects to Products & Product Management



제품 중심 개발



아키텍처 및 제품 라인 설계 ...

Base: n = 155 Gartner Research Circle Members
Q: What were or are your organization's top 3 most important reasons for adopting agile development?
On each statement, the chosen rank will show on the right of the statement. Click

© 2018 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates.

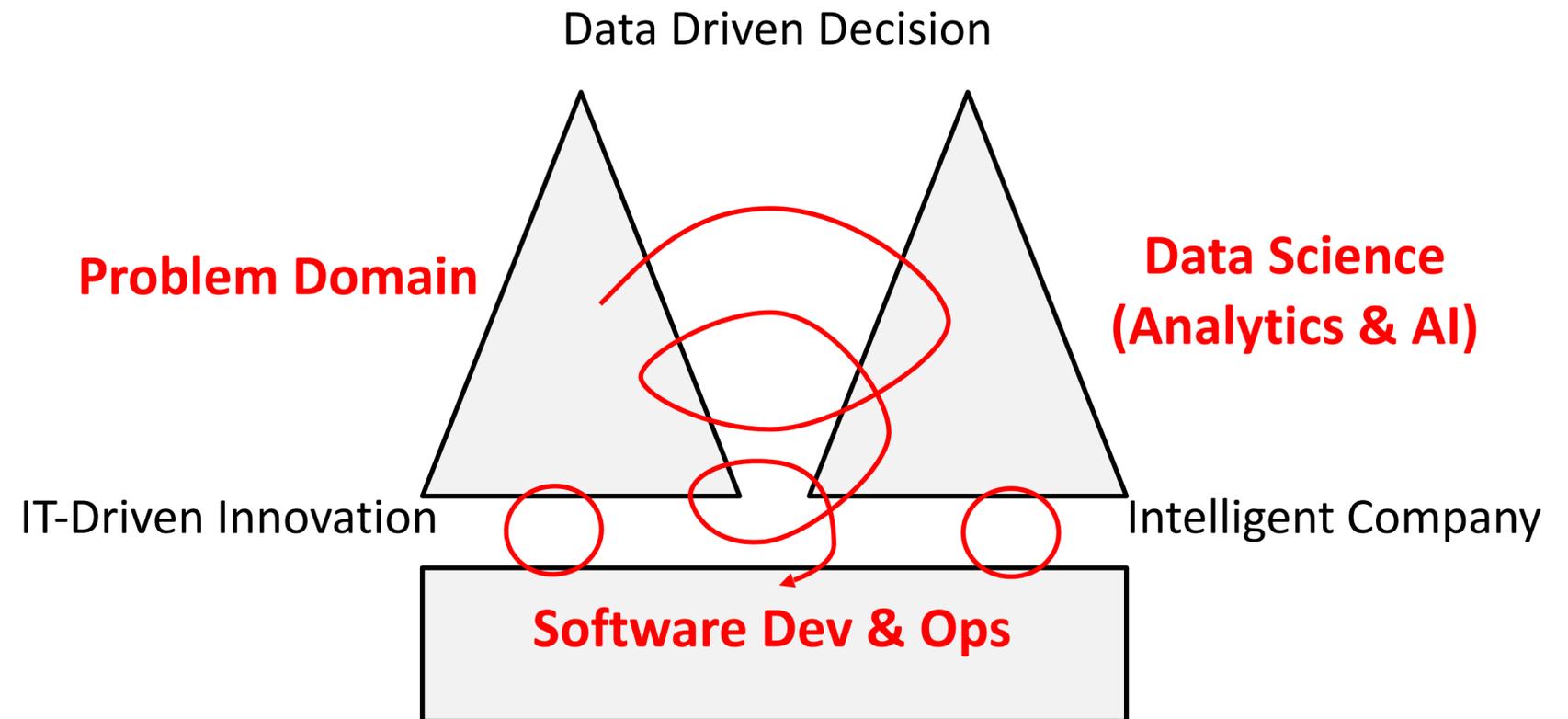
제조 분야 Digital Transformation

- **Twin Peaks (Domain & AI)**

반도체는 제조 분야 중 가장 Digitalize 되어 있음



IT 기술을 통한 혁신, 데이터 주도 의사결정 문화, 지능형 기업의 3단계 DT 전략



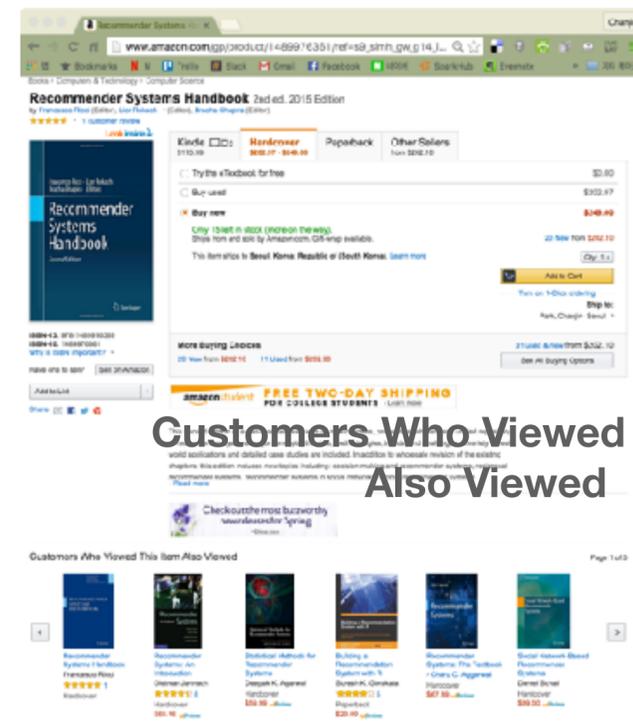
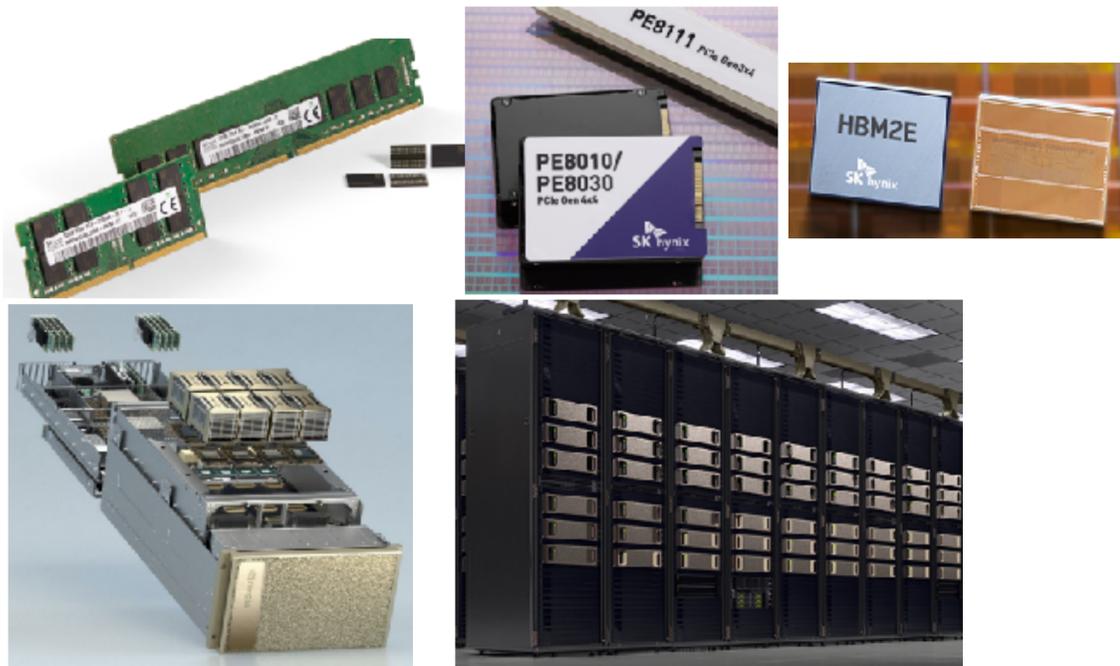
B. Nuseibeh, "Weaving together requirements and architectures," in *Computer*, vol. 34, no. 3, March 2001

데이터 사이언스가 지금 중요한 이유

- 왜 많은 기업들이 Data Science 조직을 만들고 있나?

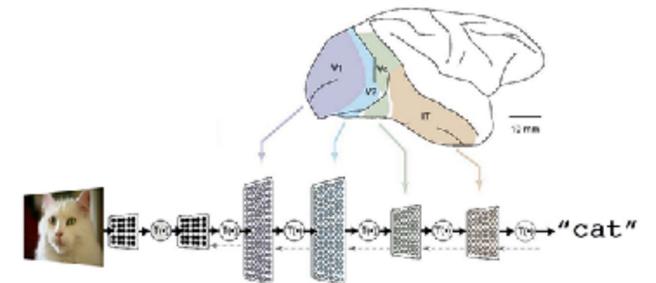
데이터를 사용하여 직접 가치를 만들고 있다

컴퓨팅 파워가 싸졌다



Customers Who Viewed This Item Also Viewed

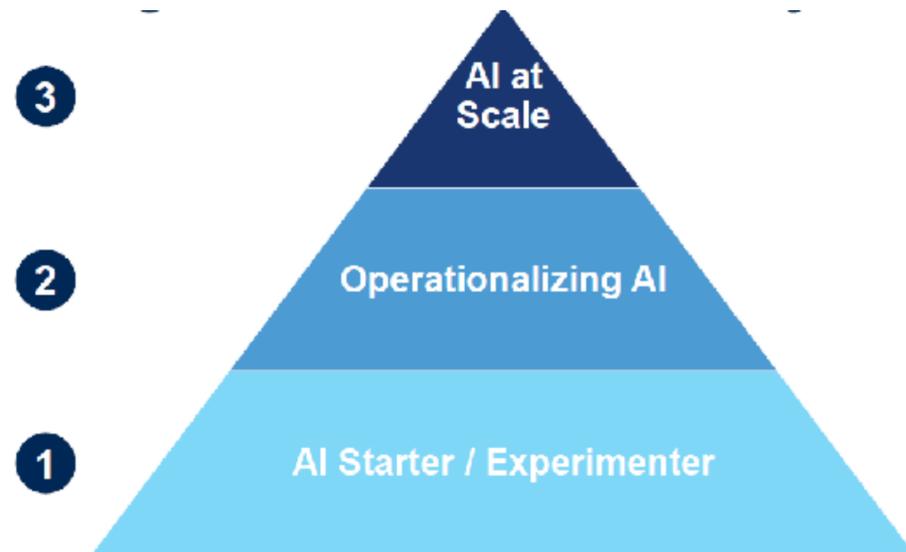
AI, 딥러닝 기술이 급속하게 발전하고 있다



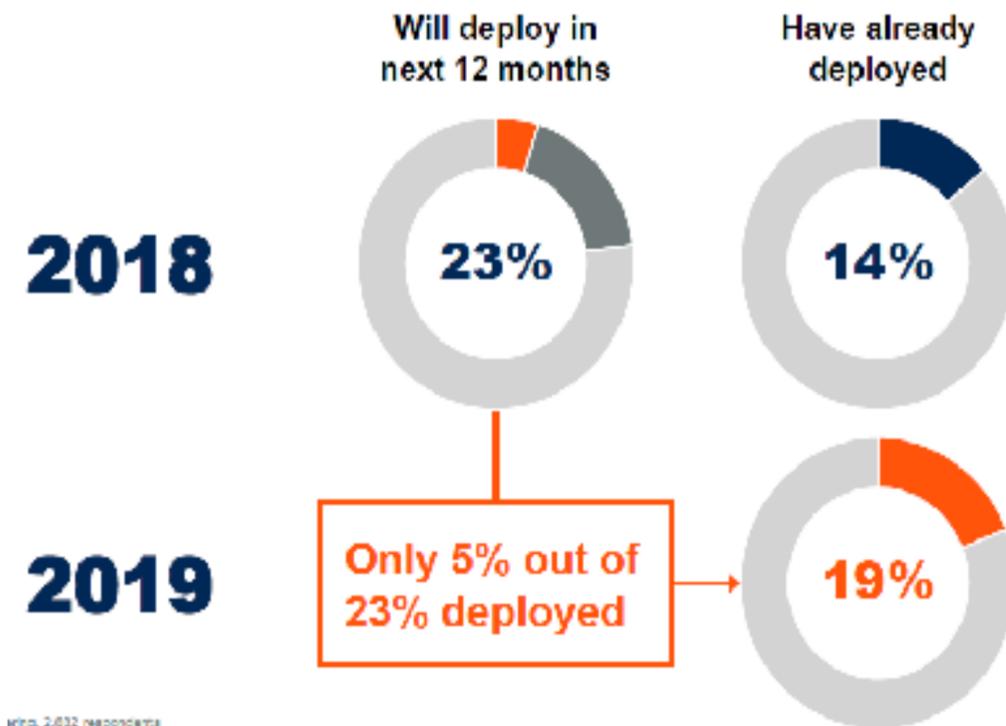
SE for Digital Transformation, 2019

- AI 리서치 즉, 실험에 초점을 두고, AI의 배포/운영 경험과 기술이 미성숙

AI 성공을 위해서는 실험, 운영, 확산의 단계를 거쳐야 한다



하지만, AI의 현장 활용 시도에 많은 실패가 있다



AI를 임베드한 End-to-end SW 개발과 운영 대응이 프로젝트 성공의 핵심

- 스킬의 부재, 데이터 품질,
- AI 잇점과 활용에 대한 이해 부족,
- AI를 기존 시스템에 통합 복잡도 등

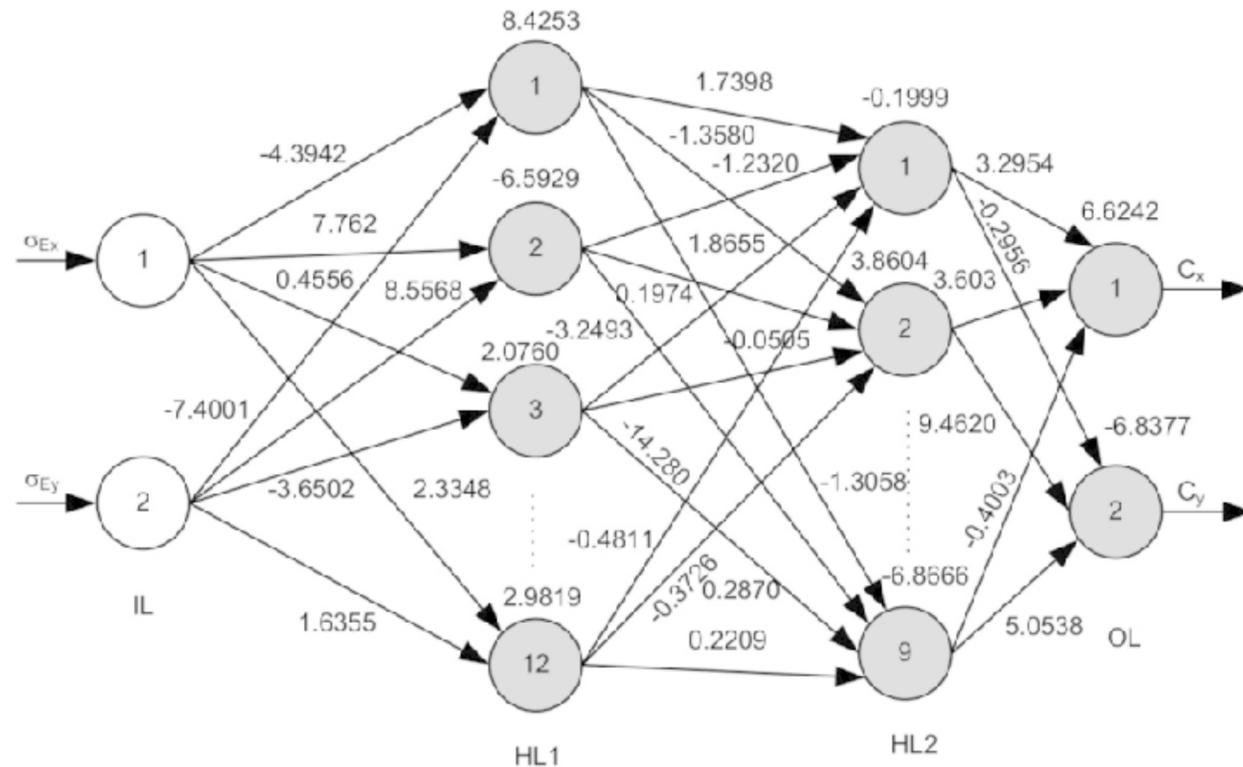
AI Engineering

AI is a New Type of Software

- **Software 2.0 by Andrej Karpathy, 2017**

AI는 새로운 형태의 SW

Neural networks are not just another classifier, they represent the beginning of a fundamental shift in how we write software. They are Software 2.0.



<https://medium.com/@karpathy/software-2-0-a64152b37c35>



Andrej Karpathy
@karpathy

Gradient descent can write code better than you. I'm sorry.

3:56 PM - 4 Aug 2017

343 Retweets 1,161 Likes



72 343 1.2K



Add another Tweet



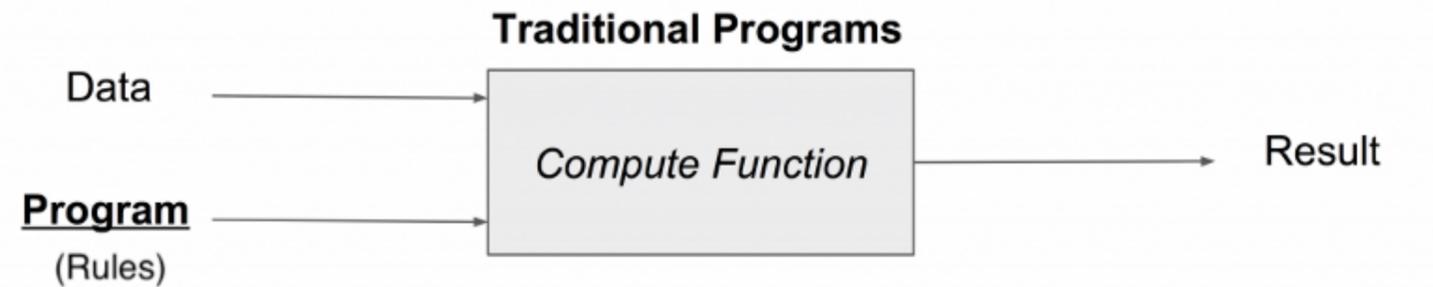
David Pfau @pfau · 5 Aug 2017
Replying to @karpathy



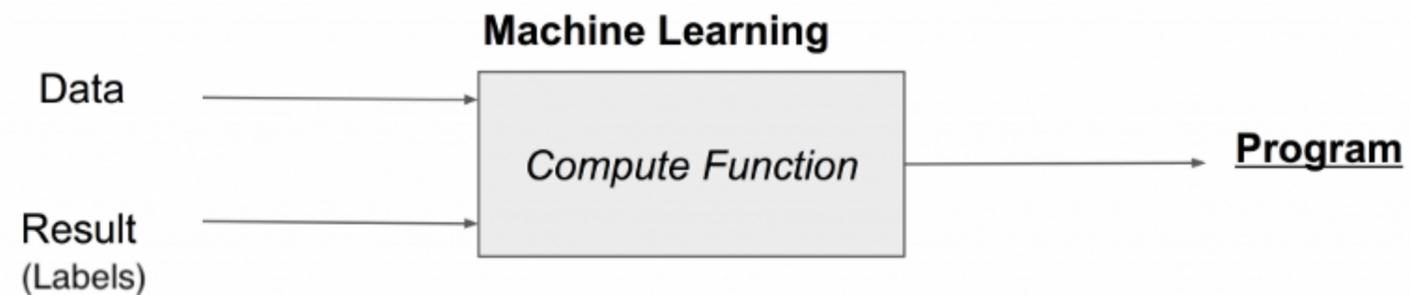
18

Difference between AI and Coding

- AI는 데이터로 만들어진 SW



기존 프로그래밍은 주어진 데이터에 적절한 결과를 얻기 위해, 프로그래머가 컴퓨터의 동작을 명령하는 코드를 작성하는 것 (Imperative)



AI와 ML은 데이터와 Label을 입력으로 학습하여 프로그램을 얻는 과정

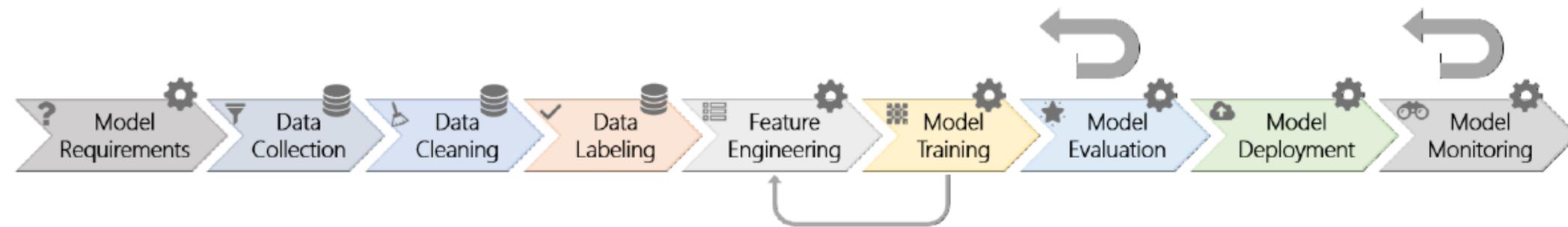
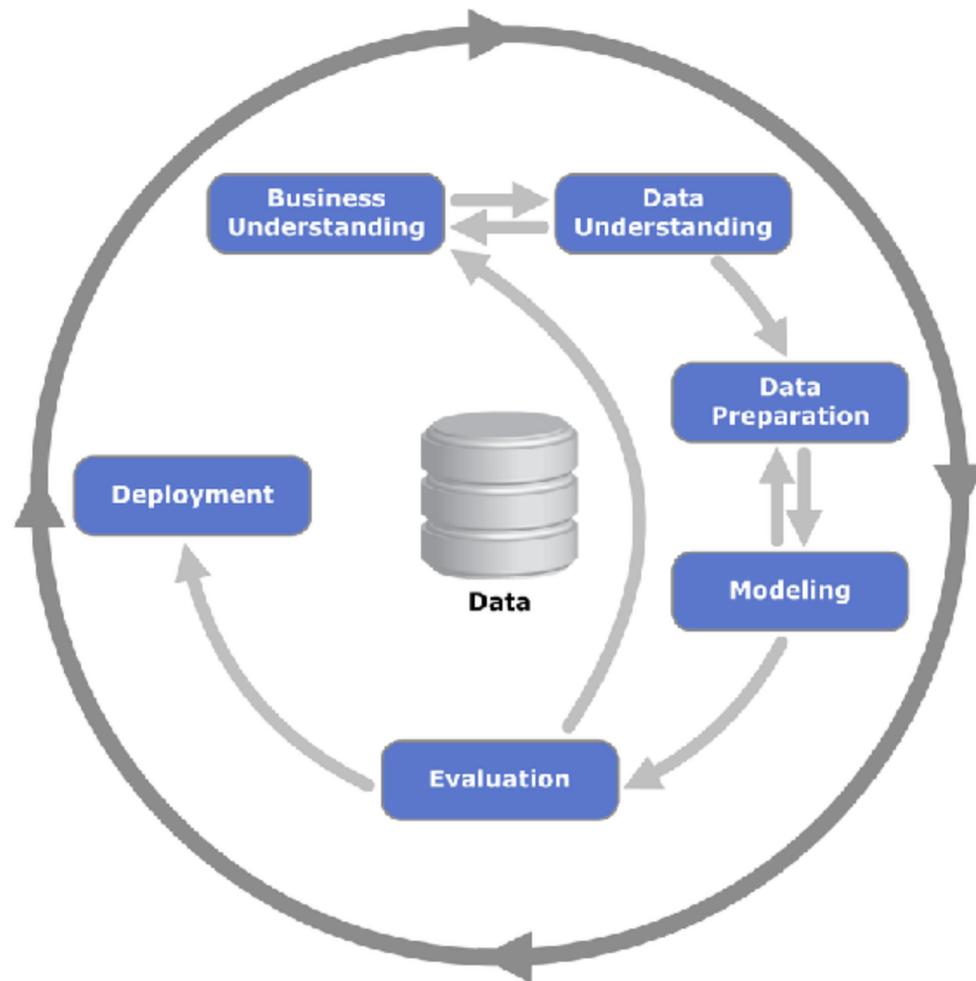
Compiling == Optimization?

The 2.0 programmers manually curate, maintain, massage, clean and label datasets; each labeled example literally programs the final system because the dataset gets compiled into Software 2.0 code via the optimization.

- Andrej Karpathy, 2017

How to Develop AI model?

- 기존 AI 프로세스는 주로 모형 개발에 초점을 두고 있음



Saleema Amershi, et. al., "Software engineering for machine learning: a case study", ICSE Software Engineering in Practice (ICSE-SEIP '19). 2019

- AI와 타 시스템과의 통합?
- AI 모니터링 (모델 성과와 비즈 성과)?
- AI 업데이트와 배포 체계 (AI CI/CD)?

Harper, Gavin; Stephen D. Pickett (August 2006). "Methods for mining HTS data". Drug Discovery Today. 11 (15–16).

Where AI Project is Different?

- AI 프로젝트 관리 프랙티스는 아직 성숙하지 않음



AI ML Projects		Traditional SW Projects
Data preparation > model training > model inference > model drift/tuning	Development Cycle	Design > development > debugging > production > SW maintenance > regression testing
Iteration across entire process	Development Phases	Each phase has deliverables
Infrastructure for development and product is different in principal	Infrastructure Requirement	Infrastructure for development and product similar in principal
Project lead shifts from data scientists, data engineer and then SW engineer	Project Lead	Projects lead by enterprise architects
Tools, best practices, and skills lacking and evolving	Skills and Tools	Tools, methods, and people with needed skills are available
Difficult to predict	Project Duration	Predictable
Most have limited experience with scaling AI ML on multiple projects	Organizational Maturity	Significant existing organizational experience in large scale projects



AI를 Divide & Conquer 할 수 있나? 재사용?

개발 IDE? Tensorflow, Pytorch, Jupiter, R

Unicorn to Lead? Data Scientist, Data Engineer, SW Engineer

AI를 Agile하게 개발?

성공 경험을 Reproduce하는 방법은?

AI model development is tip of iceberg in AI project

- AI 프로젝트는 End-to-end로 시스템 통합 관점에서 봐야함 (입력 데이터에서 결과의 시스템 전달)

ML-specific technical debts

데이터 수집부터 결과 피드백까지의 전체 파이프라인에 대한 설계가 필요함

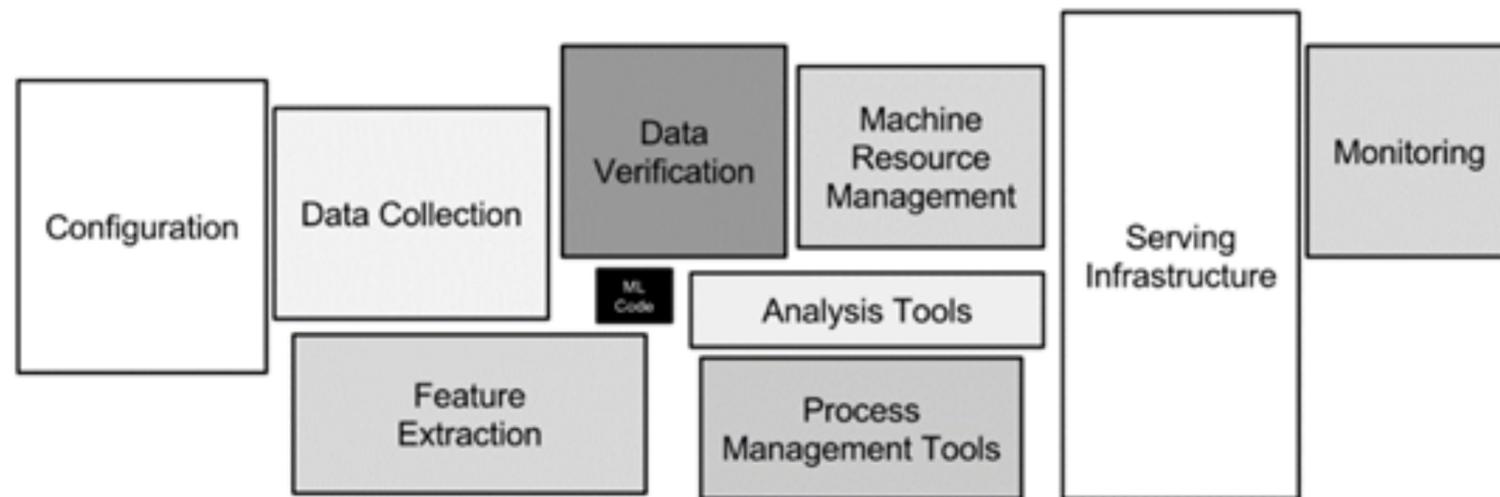


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

+ Human

AI 모니터링 및 관리라는 새로운 역할과 책임이 필요
=> 일하는 방식과 조직 변화 설계

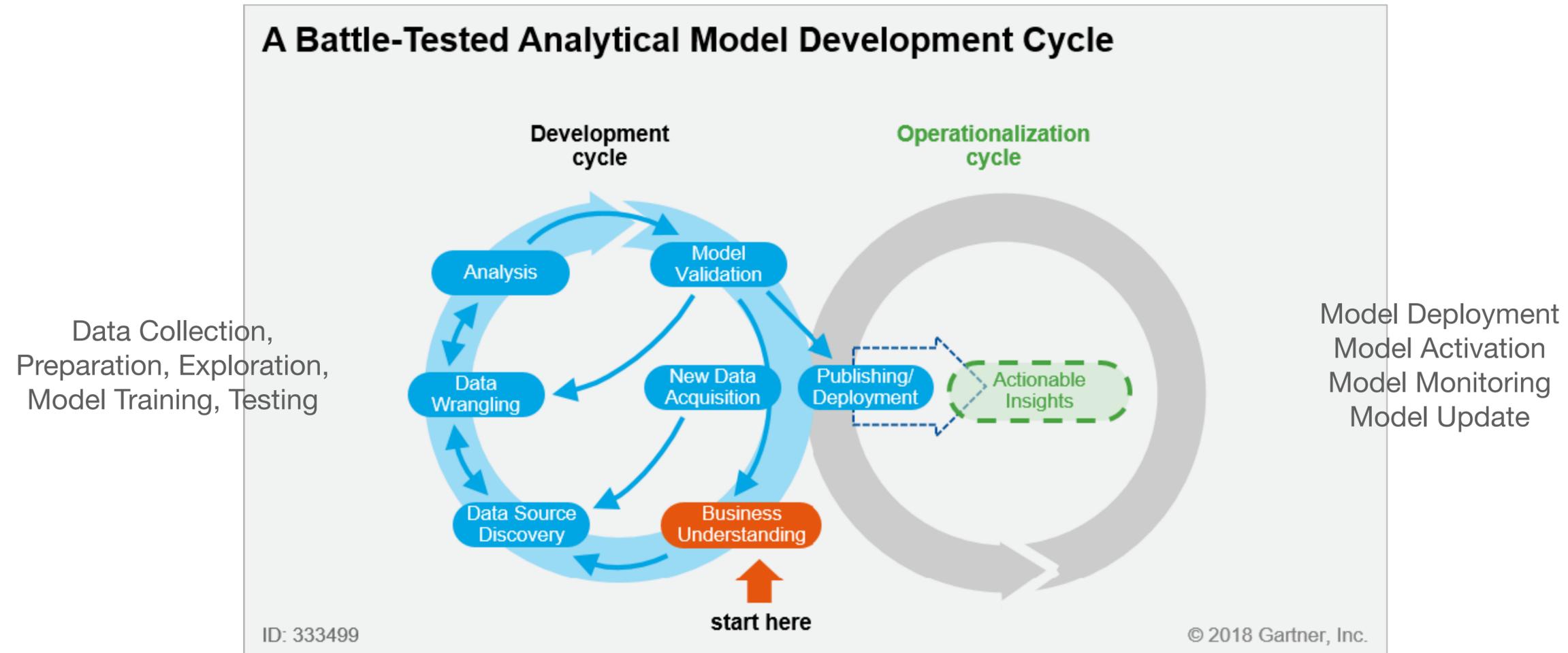
Hidden Technical Debt in Machine Learning Systems, NIPS 2015 by Google

장비로부터 불량 이미지 데이터 수집부터 결과의 MES로의 전달까지 설계. GPU 자원과 AI 개발 환경, 모델 배포 및 수행 환경 모델의 성과 모니터링, 불량 리뷰 및 자가 모델 업데이트 등

DS Projects Need to be Extended to Operationalization

- AI 모델 개발과 운영시스템화를 고려해야 함 (가트너)

AI operationalization is last mile toward business value



현재 AI는 개발과 릴리스, Publish에 중점을 둔
기술 우선으로 생각하며, 비즈니스 영향 및 운영시스템화에 대한 고려가 부족

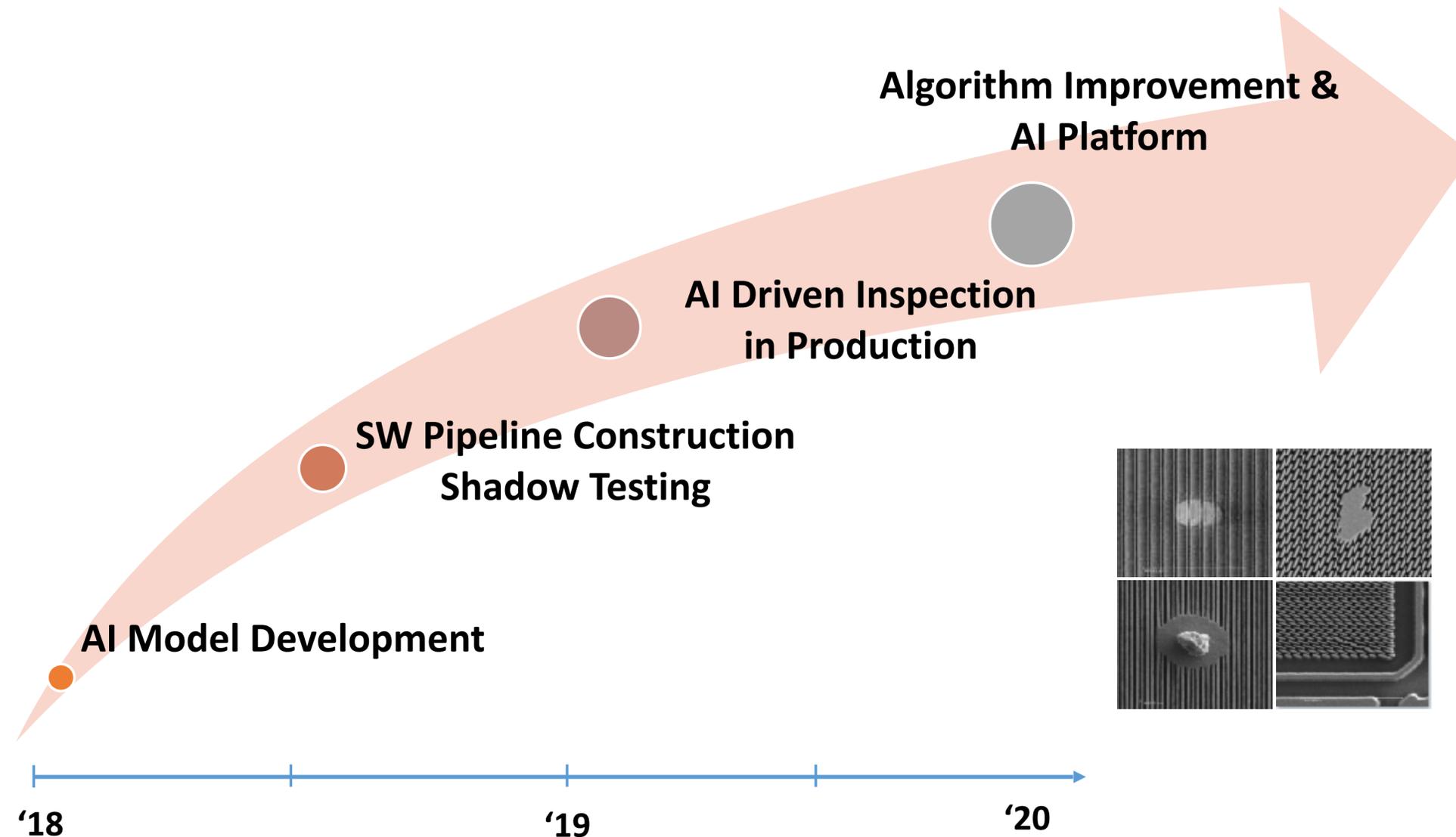
AI Project Case

(Intelligent Visual Inspection Analytics)

AI Project for Automated Visual Inspection

- **IVIA, Intelligent Visual Inspection Analytics**

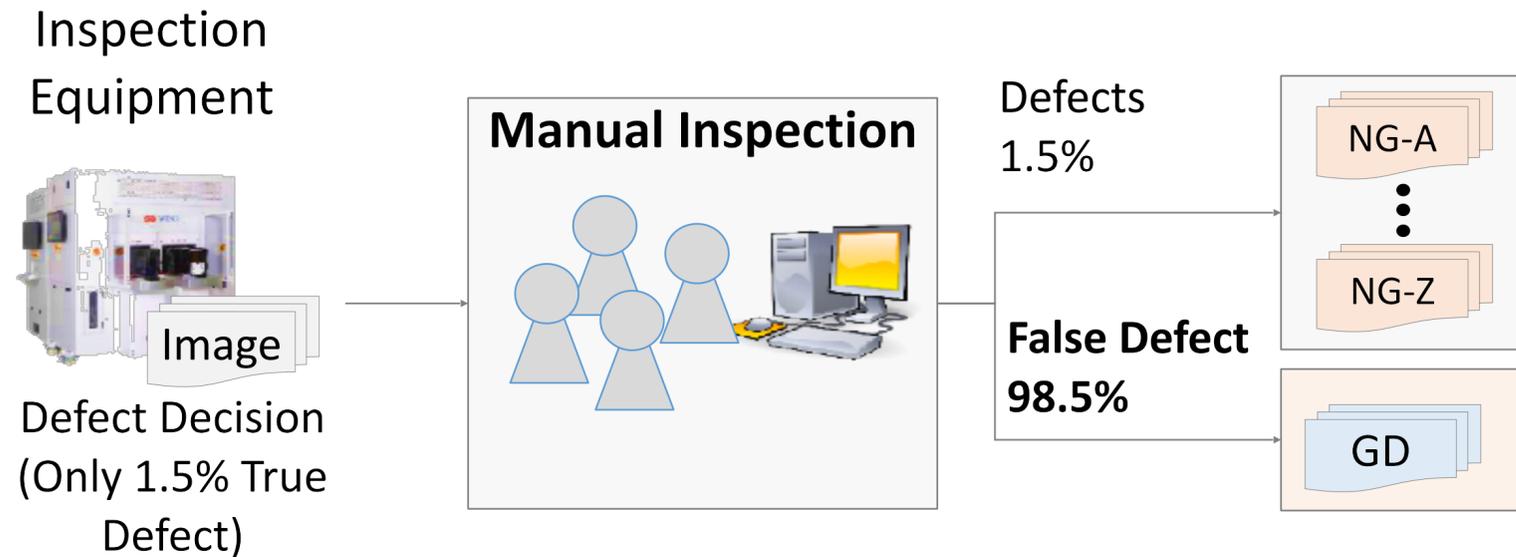
2018년에 시작. 반도체 패키징 공정에서의 수작업 불량 검사 작업을 자동화. 업무 90% 이상 감소



Problem in Semiconductor Inspection Process

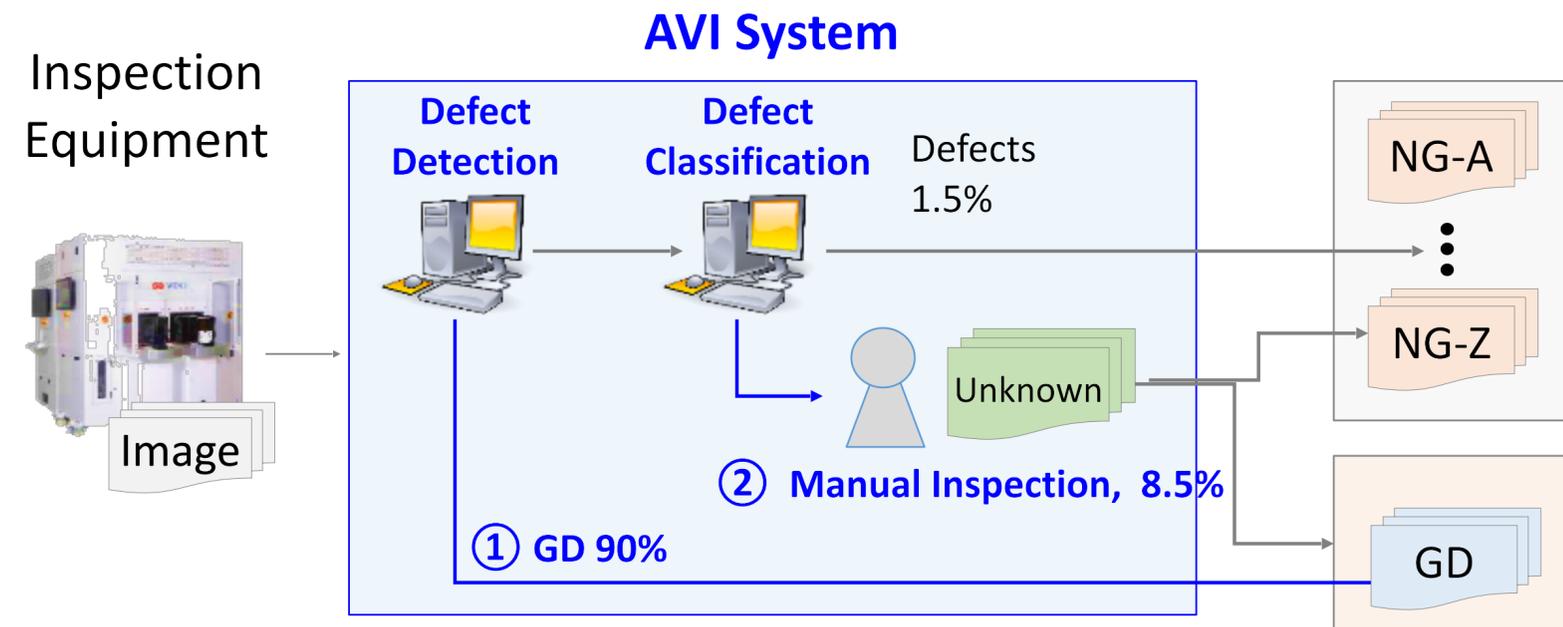
- 검사 장비의 불량 검사 성능이 낮음

검사 장비를 통해 나온 불량 후보 중 1.5% 만이 실제 불량 (False positives 98.5%)



작업자가 눈으로 보고 실제 불량을 판별. 과중한 반복적 수작업 검사가 필요
불량 정의 규칙이 복잡하기 때문에, 장비 업체는 가능한한 안전한 결과를 리포팅함

90%의 양품을 자동으로 걸러내고, 불량 분류가 애매한 8.5%에 대해 수작업 검사 (AI 성능 모니터링)

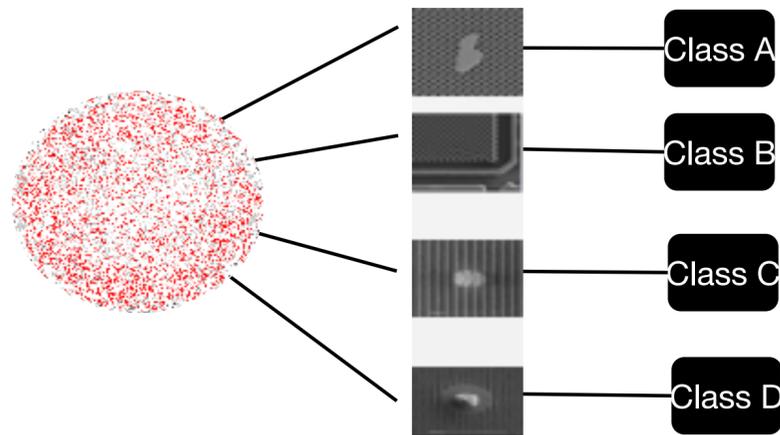


Applying Deep Learning to Visual Inspection

- 딥러닝은 이미지와 레이블 데이터만으로 사람 개입 없이 학습 (End-to-end 학습)

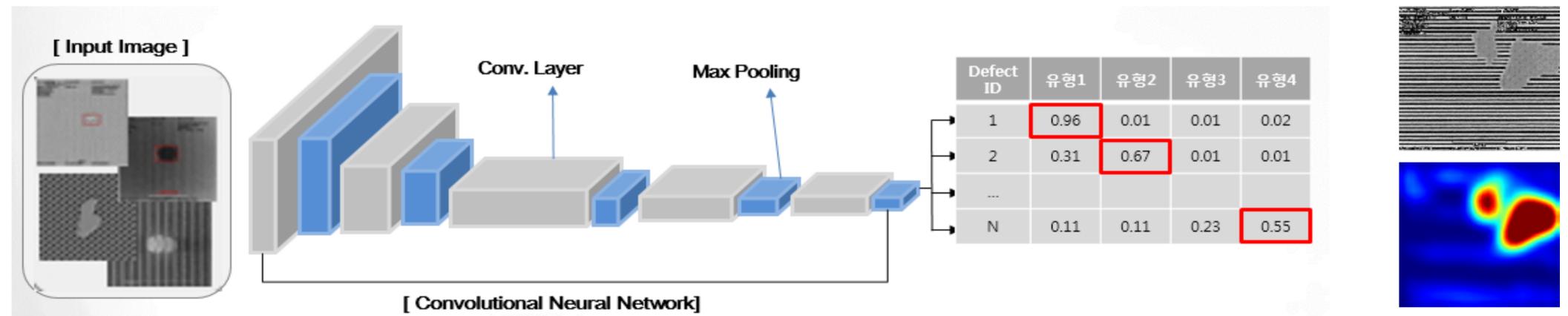
비전 기술 기반 머신 러닝 접근

정확도가 낮음. 불량을 정의하기에 부족한 비전 특성 (Features, 모양, 크기, 그림자 등)
변화에 취약하고 비전 전문가 필요 (카메라의 조도나 촬영 각도 변화, 불량 정의 규칙 변화, 새로운 불량 유형 출현 등)



< Shape, Size, Shade and etc. >

Training Efforts for changes ↑
Difficult to scale out and manage



딥러닝 접근

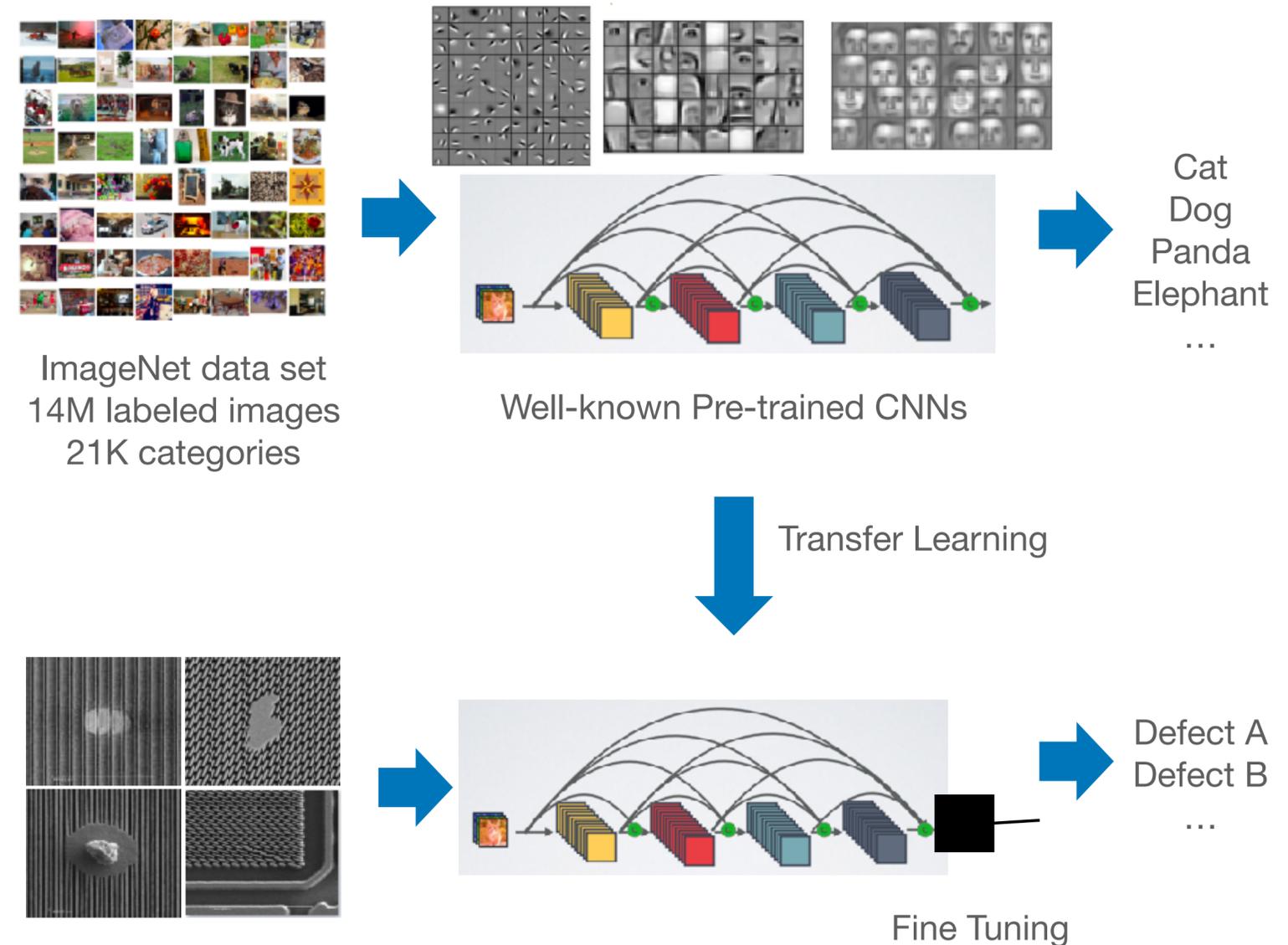
정확도 대폭 개선 (70% to 95%). 학습에 이미지 전문가가 불필요. 빠른 변화 대응. 신규 제품에 모델 셋업 시간 단축

Feasibility Study

- **실현가능성 검토 (Feasibility study)**
 - 문제가 AI로 풀 수 있는 문제인지 확인
- **프로젝트 초기에는 레이블된 데이터 부족**
 - 불량 이미지와 분류값이 있는 데이터 셋 확보가 관건
 - 레이블링 작업은 주로 수작업 (데이터 품질 문제)
 - 불량 타입 당 500개 이상의 레이블된 이미지 필요 (ImageNet)
- **불량 정의 규칙을 이해하려면 현장 엔지니어와의 협업 필수적임**

전이 학습 사용 (Transfer Learning)

잘 알려진 미리 학습된 DL 네트워크를 가져와서 마지막 판단 층만 불량 데이터로 학습 시킴 (Resnet, EfficientNet 등)



정량적 과제 목표가 정의되어야 함

- 개발 초기 AI 목표 설정이 중요하다

많은 사람들이 AI에 대해 높은 기대를 가지고 있음

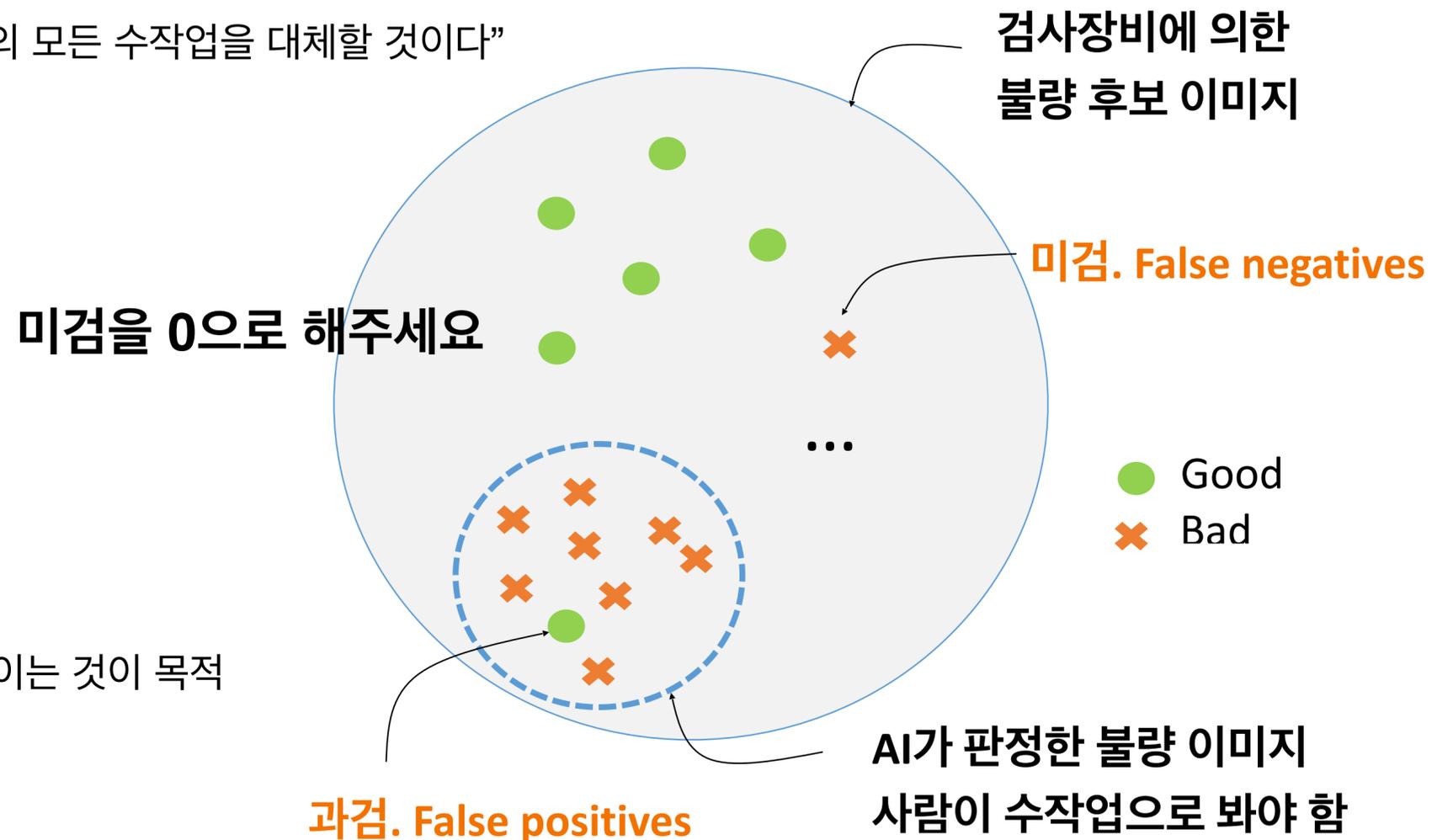
“AI는 사람이 잘 발견하기 어려운 불량도 탐지할 수 있고, 머지않아 사람의 모든 수작업을 대체할 것이다”
불가능한 목표를 제시

숫자에 기반한 목표 합의가 중요

Metric이 정의되어야 함: 과검, 미검, 정확도
현재의 검사 성능을 측정해야 AI 성능과 비교할 수 있음
그 동안 사람의 불량 검사 정확도는?

비즈니스 목표는 다를 수 있음

공정의 앞 단계는 생산성이 중요하고, 뒤 단계에서는 품질이 중요
즉 앞단계에서는 과검을 줄이는 것이 목적이고, 뒤 단계에서는 미검을 줄이는 것이 목적



AI 개발에 도메인 이해가 중요

- 불량 정의 규칙은 복잡하고, 현장엔지니어와의 협업이 중요

먼지가 떨어진 위치에 따라 불량일 수도 아닐 수도 있음

불량 이미지 형태에 대한 이해와 불량 판정 규칙을 이해해야 함

Object Detection Layer를 두어 계층적 네트워크 구성 필요 (U-net)

이미지 크기가 1K x 1K 픽셀이지만, 어떤 불량은 10 x 10 픽셀 보다 작은 크기임.

불량 영역 Segmentation 필요

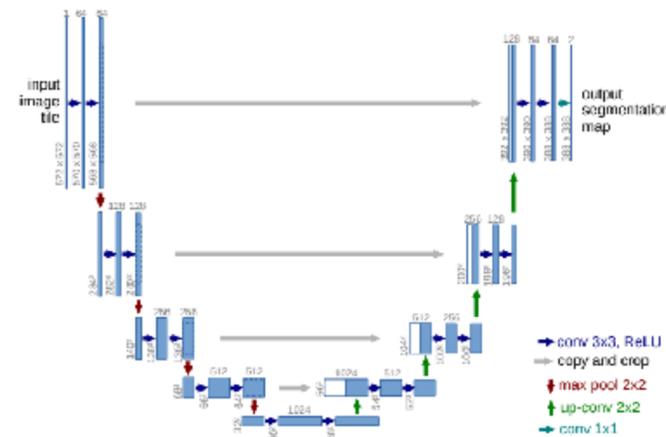
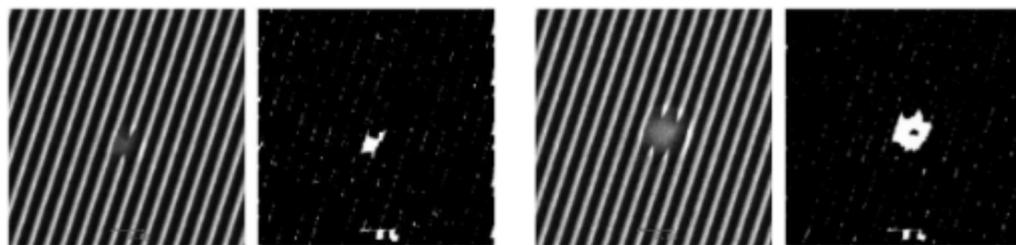
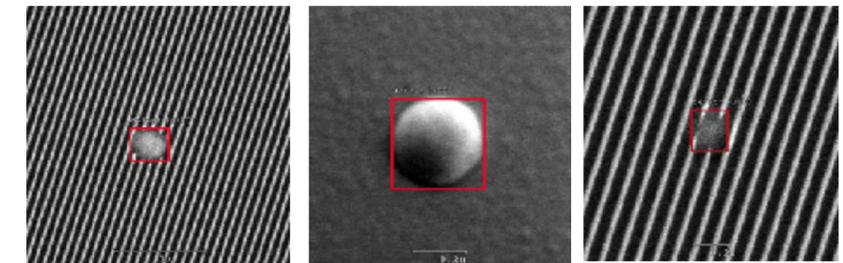


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

백그라운드 이미지 처리

제품 별로 백그라운드 이미지가 다르므로 전체 리 되어야 함

배경을 사전 처리하거나 배경 별 모델을 구성



Landing AI와의 PoC

Ronneberger O., Fischer P., and Brox T., "U-Net: Convolutional Networks for Biomedical Image Segmentation", MICCAI 2015

MVP Development

- AI 기능 검증을 위한 최소 SW 파이프라인의 구축 및 통합이 필요함

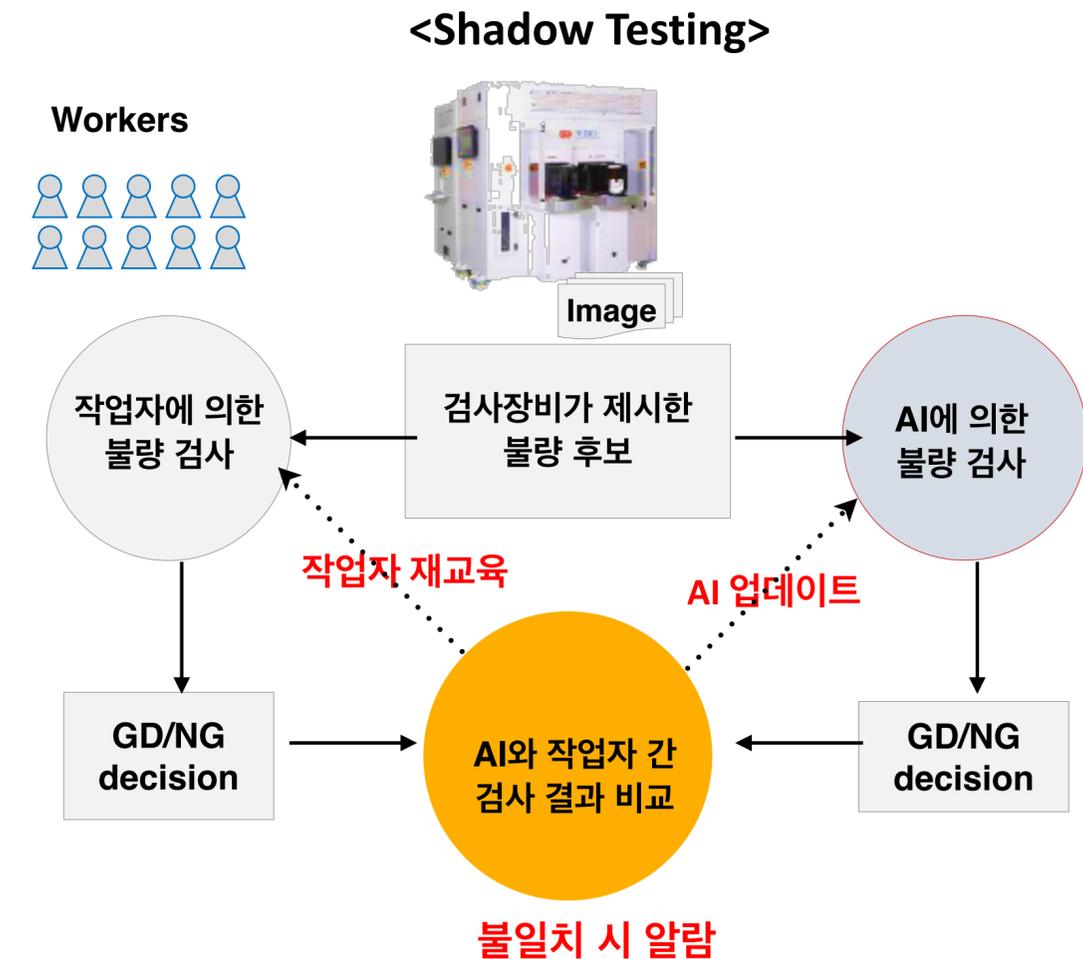
MVP 개발 = AI를 임베드한 End-to-end SW 파이프라인 구축

데이터 수집 부터, AI 판단, 결과 전달, 결과 확인을 위한 UI 까지의 개발



AI 신뢰 확보를 위한 Shadow 테스트

AI와 사람의 병렬 검사. 정확도 비교. AI 개선 혹은 작업자 재교육

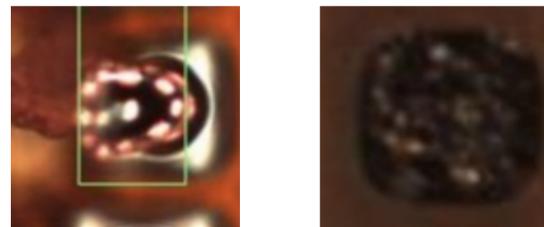


현장 적용을 위한 문제 해결

- 불량 분류 AI를 만든 후 현장 적용을 위해서는 다양한 문제 해결이 필요하다

새로운 타입의 불량 출현으로 인한 AI 학습

적용 장비 확대로 인한 신규 불량
제조 공정 개선 및 변화로 인한 신규 불량

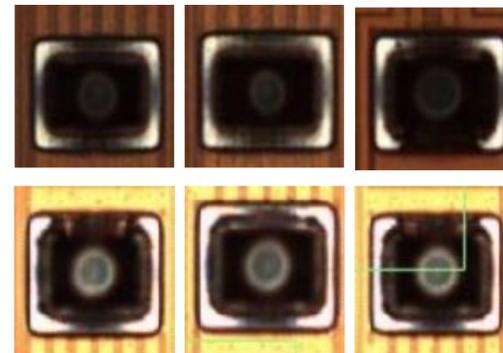


Abnormal bump

?

장비 상태 변화로 인한 AI 학습

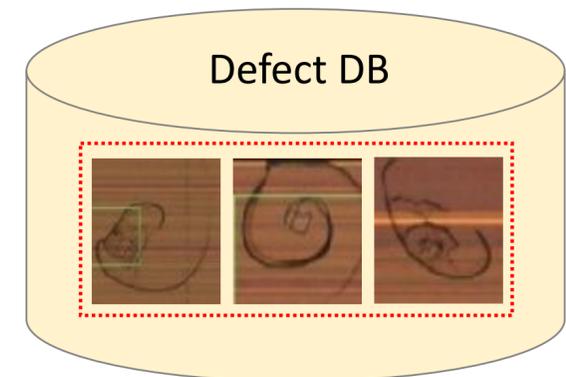
조도의 변화, 촬영 각도 변화, 장비 노후화로 인한
AI 성능 저하



이미지 밝기/명암 변경

불량 정의 규칙의 변화로 인한 AI 모델 업데이트

Ground Truth의 변화

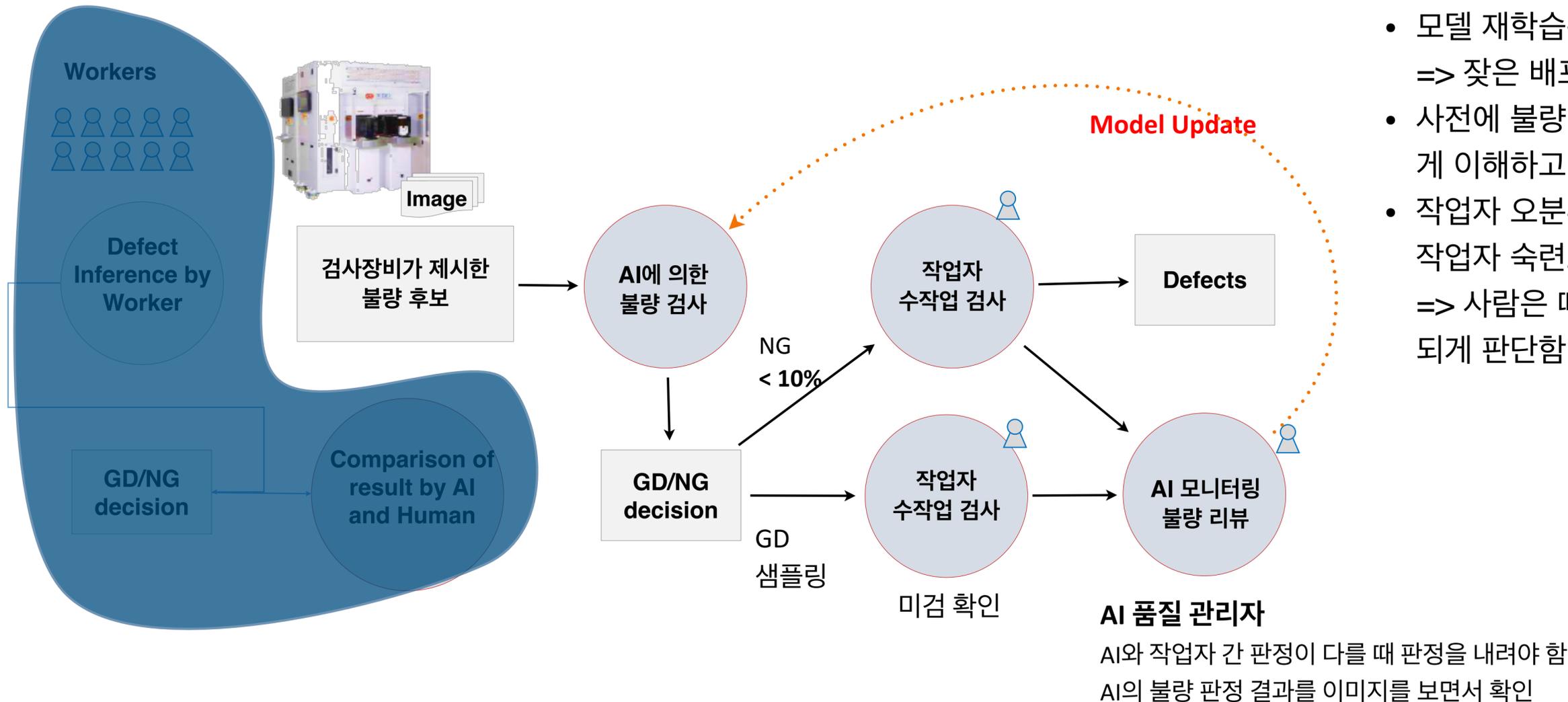


Thread Shape Defect Will be
Treated as Good Product

MVP Development & Shadow Testing

- AI 성능이 저하 시 작업자와 리뷰 협업이 중요 (Postmortem walkthrough)

AI 주도 검사체계로 전환



Lessons Learned

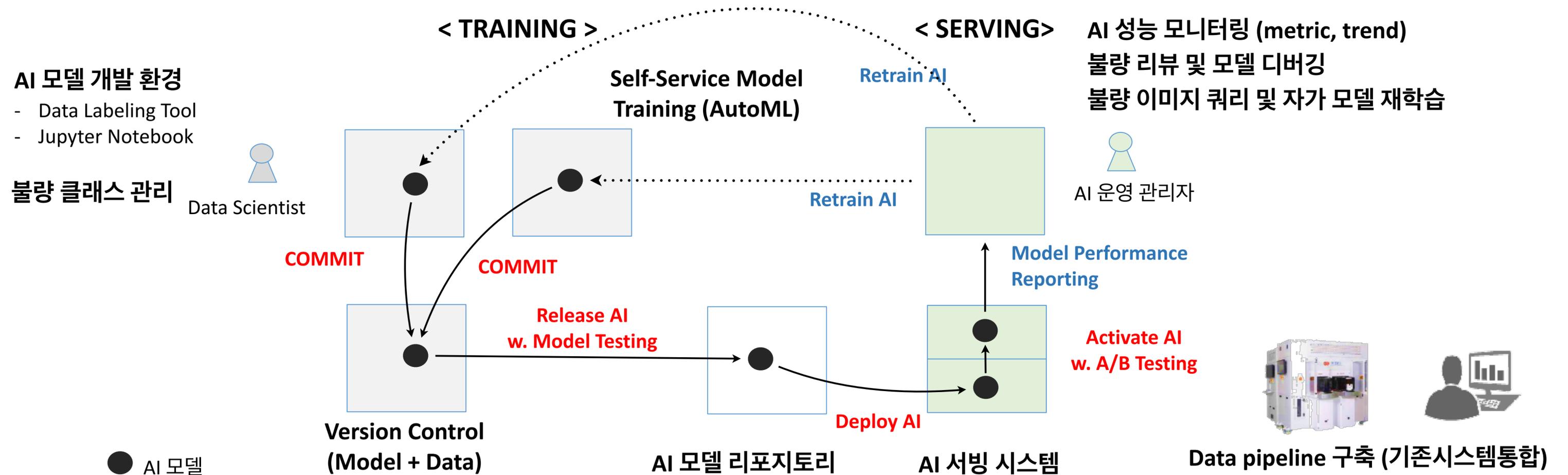
- 모델 재학습은 다양한 이유로 수시로 일어남
=> 잦은 배포가 가능한 체계 구축 필요
- 사전에 불량 규칙의 재정의 작업이 필요했음. 작업자마다 다르게 이해하고 있는 경우가 있음
- 작업자 오분류 다수 발생. 작업자 판단이 항상 옳은 것은 아님. 작업자 숙련도가 다르고, 한 작업자도 결과가 일관되지 않음
=> 사람은 때때로 실수를 하며 일관되지 않은 반면, AI는 일관되게 판단함

Production

- **IVIA를 다른 장비/제품으로 확대 적용 (Scale out IVIA)**

- IVIA를 전 검사 장비와 다른 제품으로 확대. 장비 간 차이, 제품 간 차이로 성능이 저하됨. 일반화와 특화 모델 작업

제품 중심의 플랫폼 접근 필요 - 학습과 운영을 위한 AI 모델 흐름



AI Project Phases

- AI 프로젝트의 개발 단계



- 실현 가능한 목표 설정 중요
- 학습 데이터 셋 구축 필요
- 도메인 지식이 AI에 반영되어야 함
현장엔지니어 협업 중요

- AI 성능 피드백을 받을 수 있는 엔드 투 엔드 SW 파이프라인을 구축
- 입력 데이터의 변화 및 모델 성능 저하에 대응. 시스템 통합 및 Shadow 테스트를 통한 AI 및 학습 규칙 보완
- AI 모델에 의한 신뢰 확보가 관건

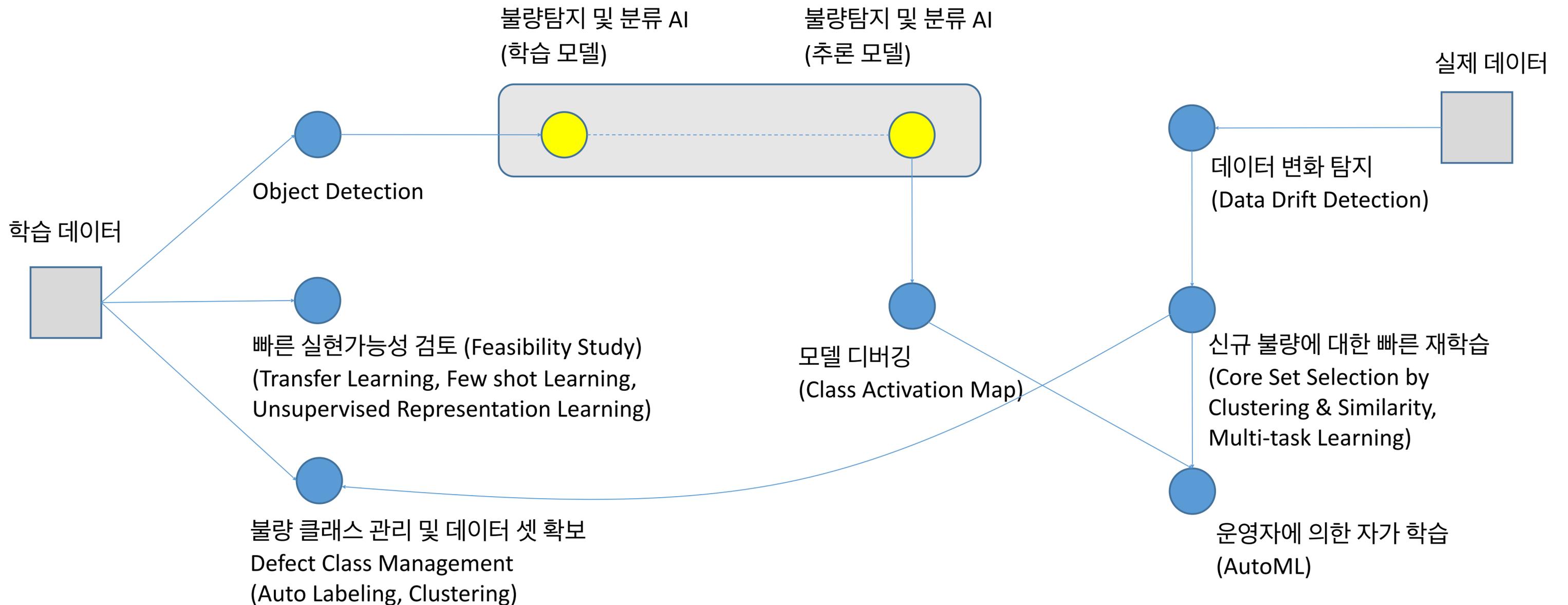
- 모델 성능 저하 시 백업 플랜 필요 (신속한 AI 재학습)
- AI 모델 릴리스/배포/실행 체계 구축
- 릴리스 테스트, 실행 시 A/B 테스트 필요
- AI 적용 시 새로운 일하는 방식 설계 필요



Topics of AI Engineering

Various AI Models are Needed

- 불량 탐지 및 분류 문제로 시작했지만, 실제 현장 적용을 위해서는 다양한 AI 모델 개발이 필요함



AutoML Technology

- **딥러닝 AutoML**

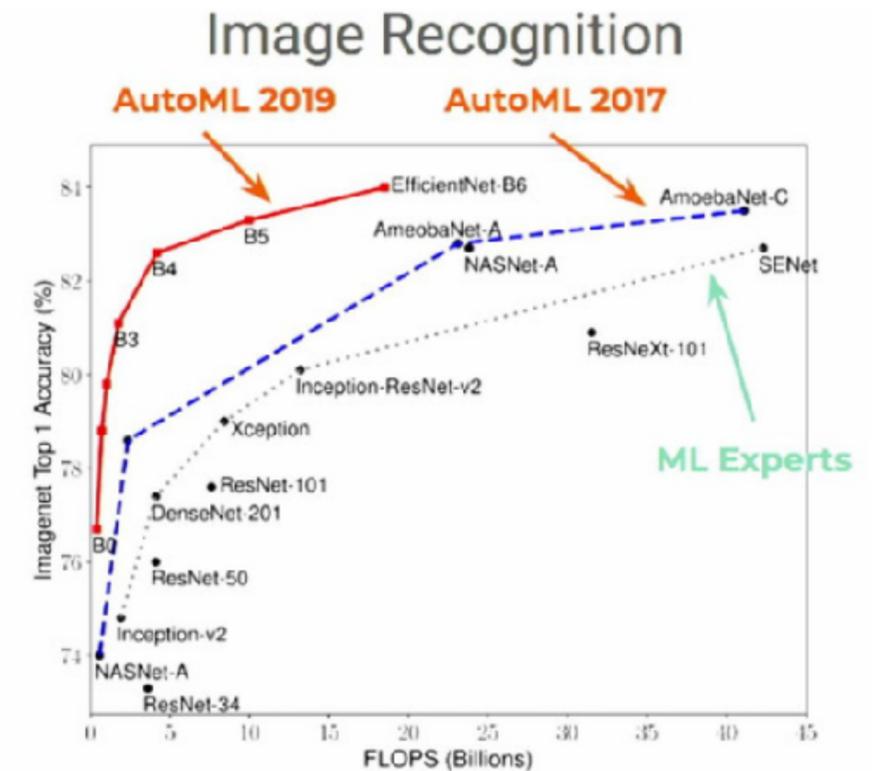
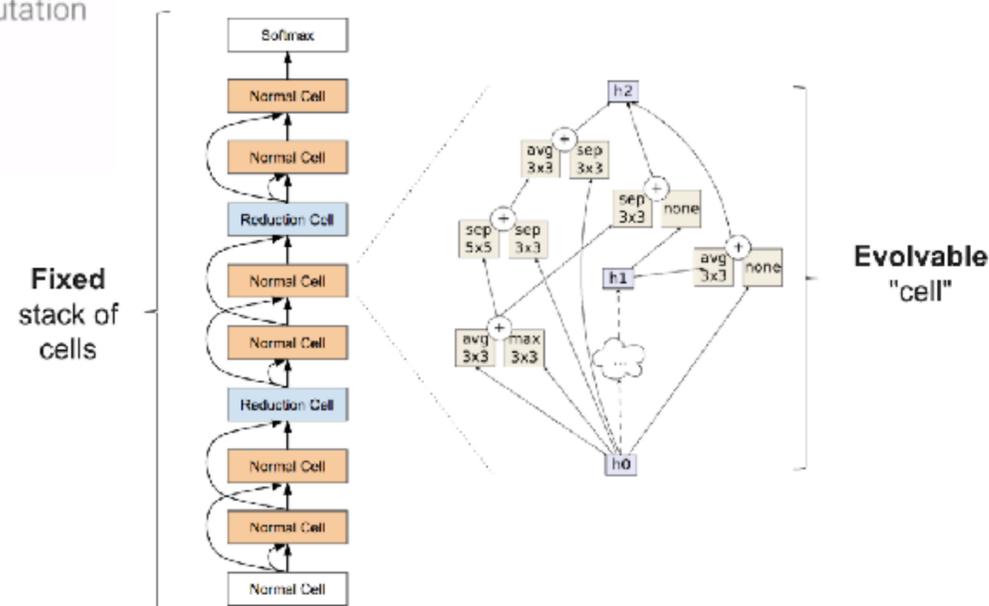
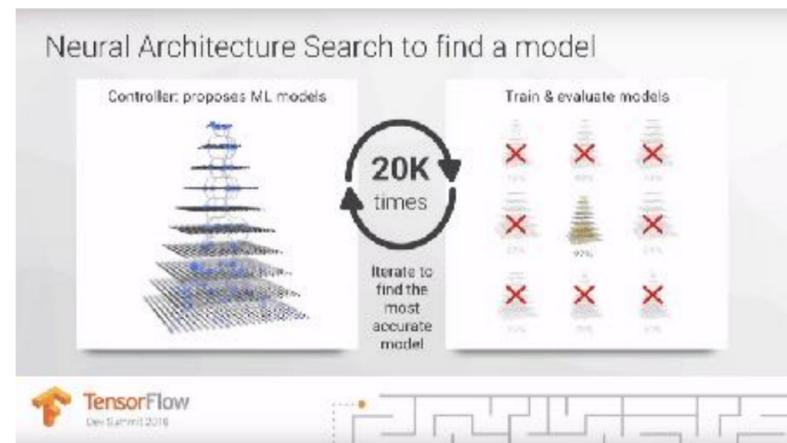
100 배의 컴퓨팅 파워로 ML 전문가를 대체 (DL without PhD)

뉴럴넷을 설계하는 뉴럴넷 (Network Architecture Search)
 Reinforcement Learning or Evolutionary Algorithms을 활용



Current:
 Solution = ML expertise + data + computation

Can we turn this into:
 Solution = data + 100X computation



Jeff Dean, Tensorflow Dev Summit 2018.

AutoML은 이미지, 동영상 등의 대부분 영역에서 전문가 설계 뉴럴넷의 성능을 앞서는 결과를 보임 (ICML 2019)
 Image Recognition, Object Detection, Language Translation, Video Classification, and etc

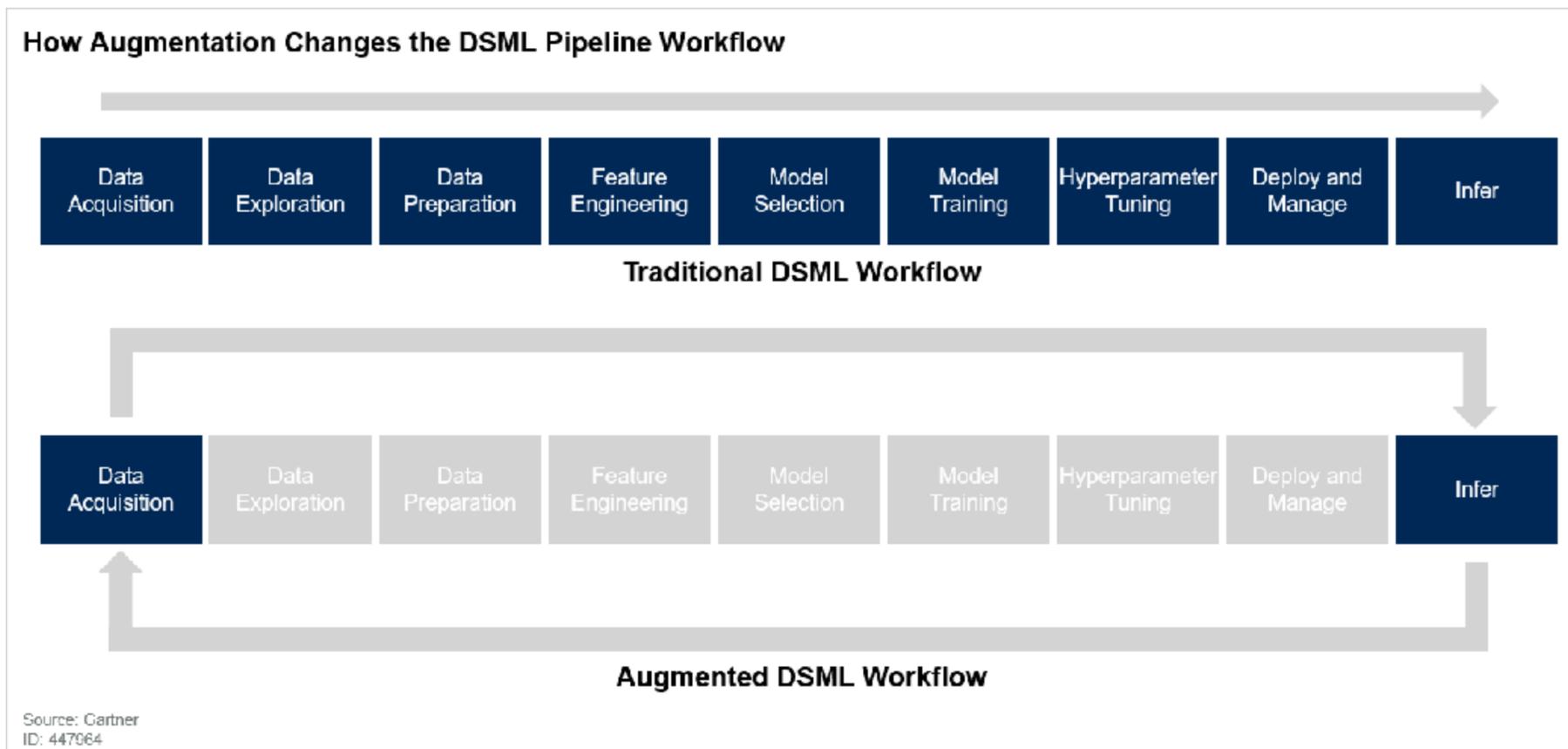
AutoML Technology

- Table 데이터 AutoML

AI 민주화 (AI Democratization)

데이터를 소유한 누구나 AI 모델을 만들 수 있다

CDS 교육 프로그램을 통해 도메인 지식을 가진 현장 엔지니어에 데이터 분석 기술을 교육시킴



현장 분석 요청에 대한 대응 효율화



DataRobot

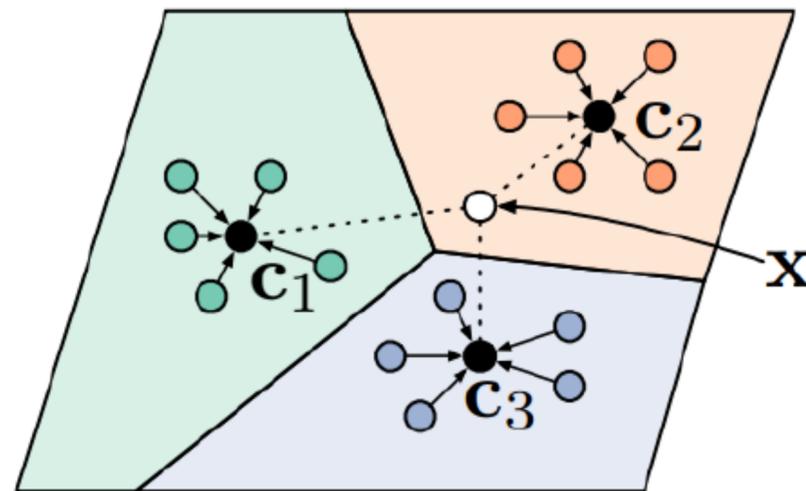


AI Feasibility Study - Few Shot Learning

- AI 개발 초기에는 데이터가 부족, 빠른 모델 개발을 위해서는 적은 수 데이터 학습 기술이 필수적임

Siamese network and prototypical network (Few Shot Learning or Meta Learning)

불량을 대표하는 레퍼런스 데이터 포인트를 가지고 뉴럴넷을 사용하여 임베딩 공간 상으로 매핑, 주어진 데이터에 대해 가장 가까운 포인트 (프로토타입)를 찾아 분류 값 설정



- Learn a non-linear mapping of the input to an embedding space using a neural network
- Classify an example by comparing the distance between the prototype representations and the example

$$P_{\theta}(y|x, C) = \text{softmax}(-L_2(f(x), c_i))$$

Snell, Jake, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning." *Advances in Neural Information Processing Systems*. 2017.

Model Evolution – Concept Drift

- 현장 적용 단계에서는 시스템 진화가 고려되어야 함

시간에 따라 동적으로 변화하는 환경에서는 (Non-stationary), 데이터 분포가 시간에 따라 바뀔 수 있음 (Concept drift)

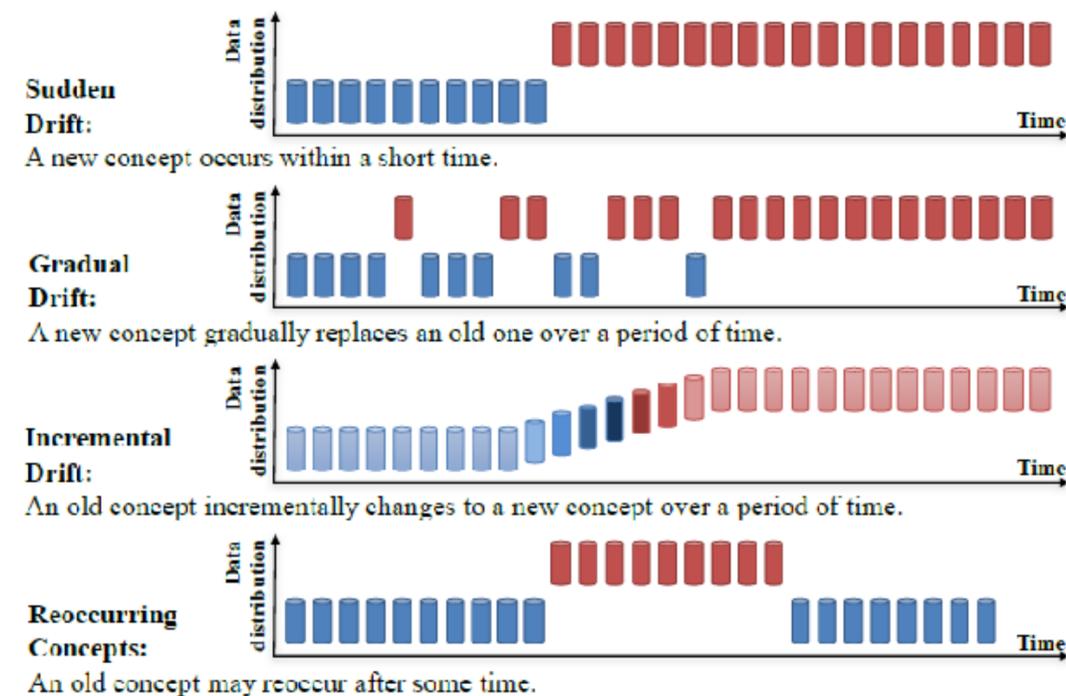


Fig. 4. An example of concept drift types.

J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE TCAD, vol. 31, no. 12, 2019.

소프트웨어 릴리스 이후 진화의 단계

모델 업데이트를 위해 데이터 사이언티스트가 필요? 네트워크 구조 설계 이미지를 모아 네트워크 가중치만 업데이트?

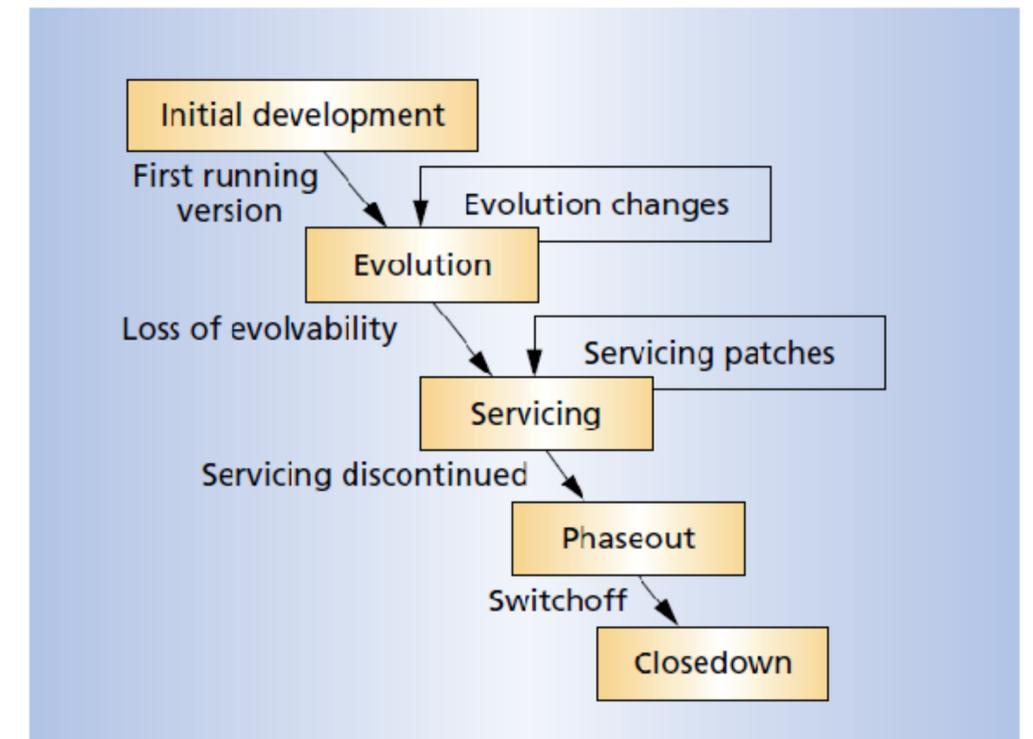


Figure 1. The simple staged model for the software life cycle consists of five distinct stages.

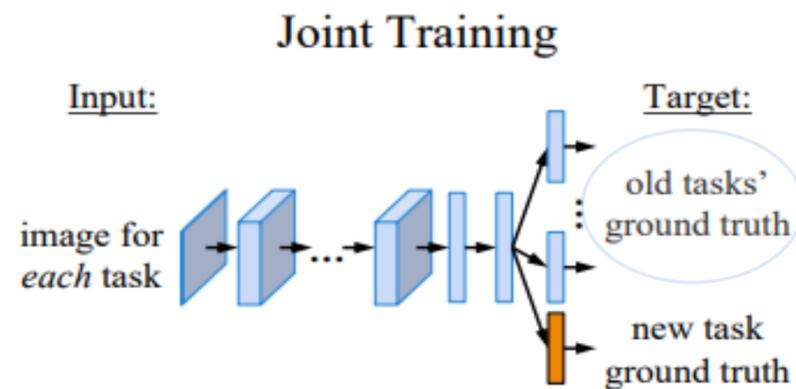
V. T. Rajlich and K. H. Bennett, "A staged model for the software life cycle," in Computer, July 2000.

Model Evolution – Learning without Forgetting

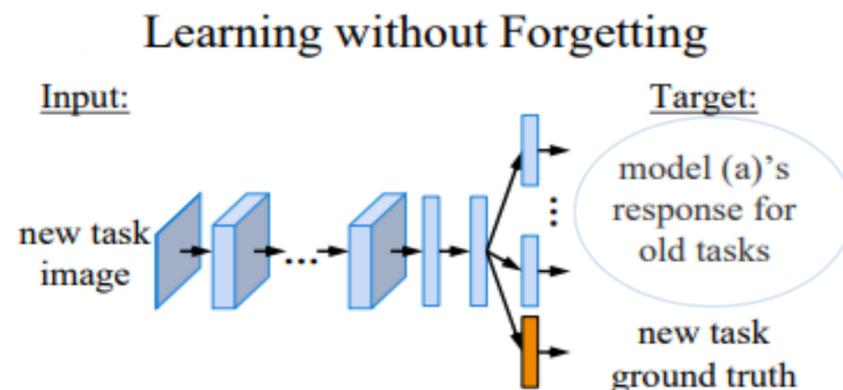
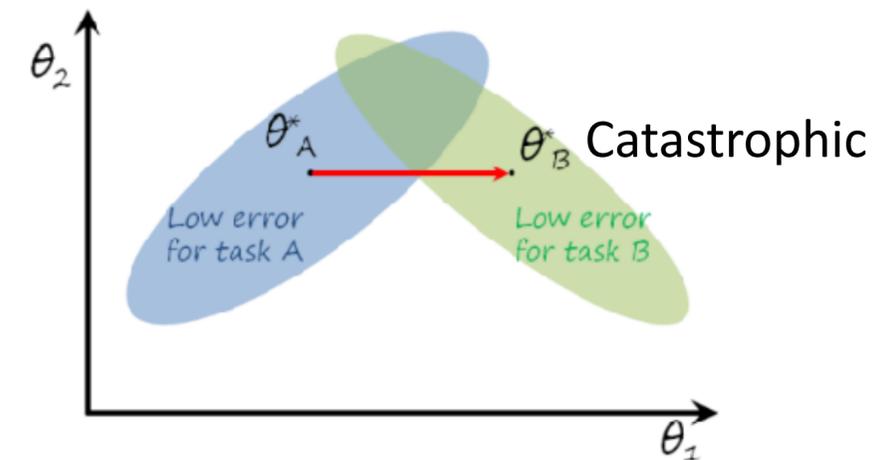
- 뉴럴넷의 잘 알려진 문제 중 하나는 새로운 사실을 기억하는 과정 속에서의 갑작스러운 망각임 (Catastrophic Forgetting, CF)
 - 뉴럴넷이 일련의 사건을 학습할 때, 새로운 사건을 학습하는 것은 이전의 사건에 대한 망각을 유발할 수 있음

Multi-task Learning (Learning without Forgetting)

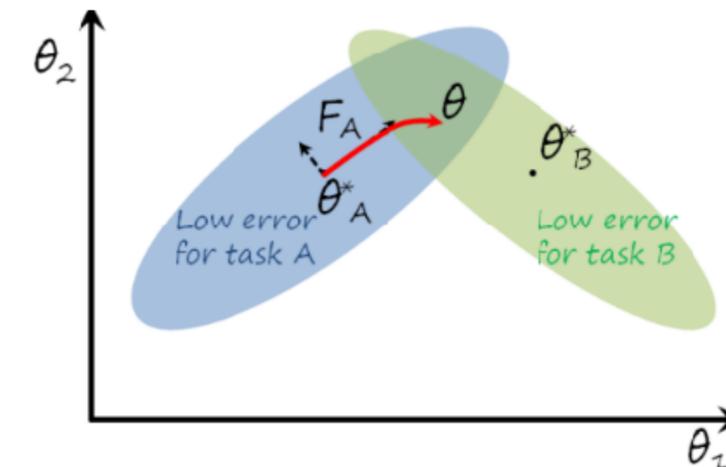
기존의 불량 탐지 및 분류 성능을 유지하면서 새로운 타입의 불량을 학습하는 방법



Jointly learning multiple tasks with old and new datasets



Using only examples for the new task, we optimize both for high accuracy for the new task and for preservation of responses on the existing tasks from the original network.



AI Model Debugging (Explainable)

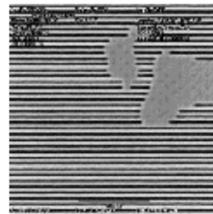
- 딥러닝의 판단 과정은 블랙박스로 알려져 있으나, 모델 신뢰를 위해서는 의사결정에 대한 해석이 필요

Class Activation Mapping

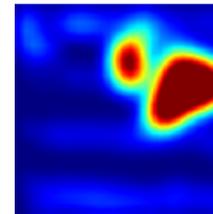


불량 분류 결과와 Activation 맵을 같이 보여주어, NN이 어느 부분을 보고 불량 판단을 했는지 알 수 있음

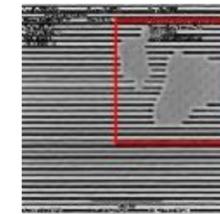
두 AI 모델이 동일 불량으로 판정했다하더라도, Activation Map을 보고 더 나은 모델을 알 수 있음



Defect Image

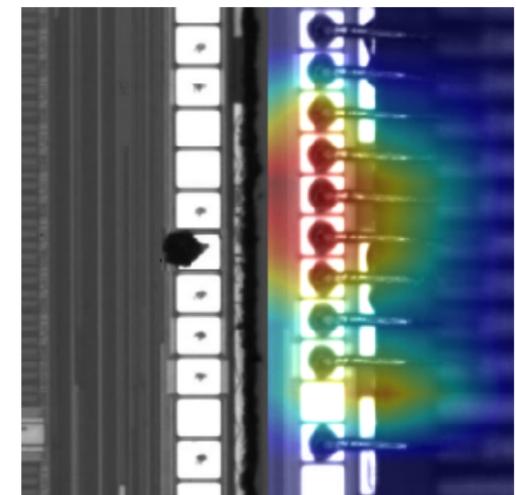
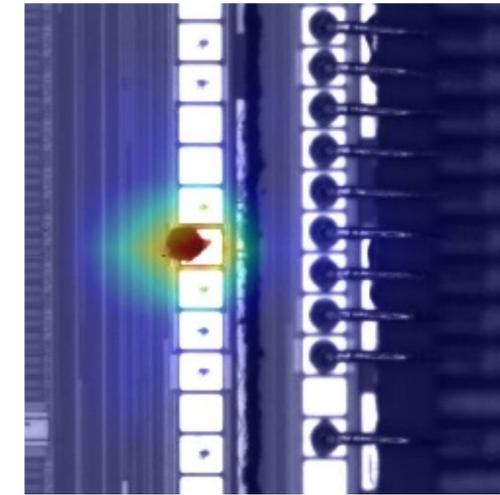


Activation Map



Object Detection

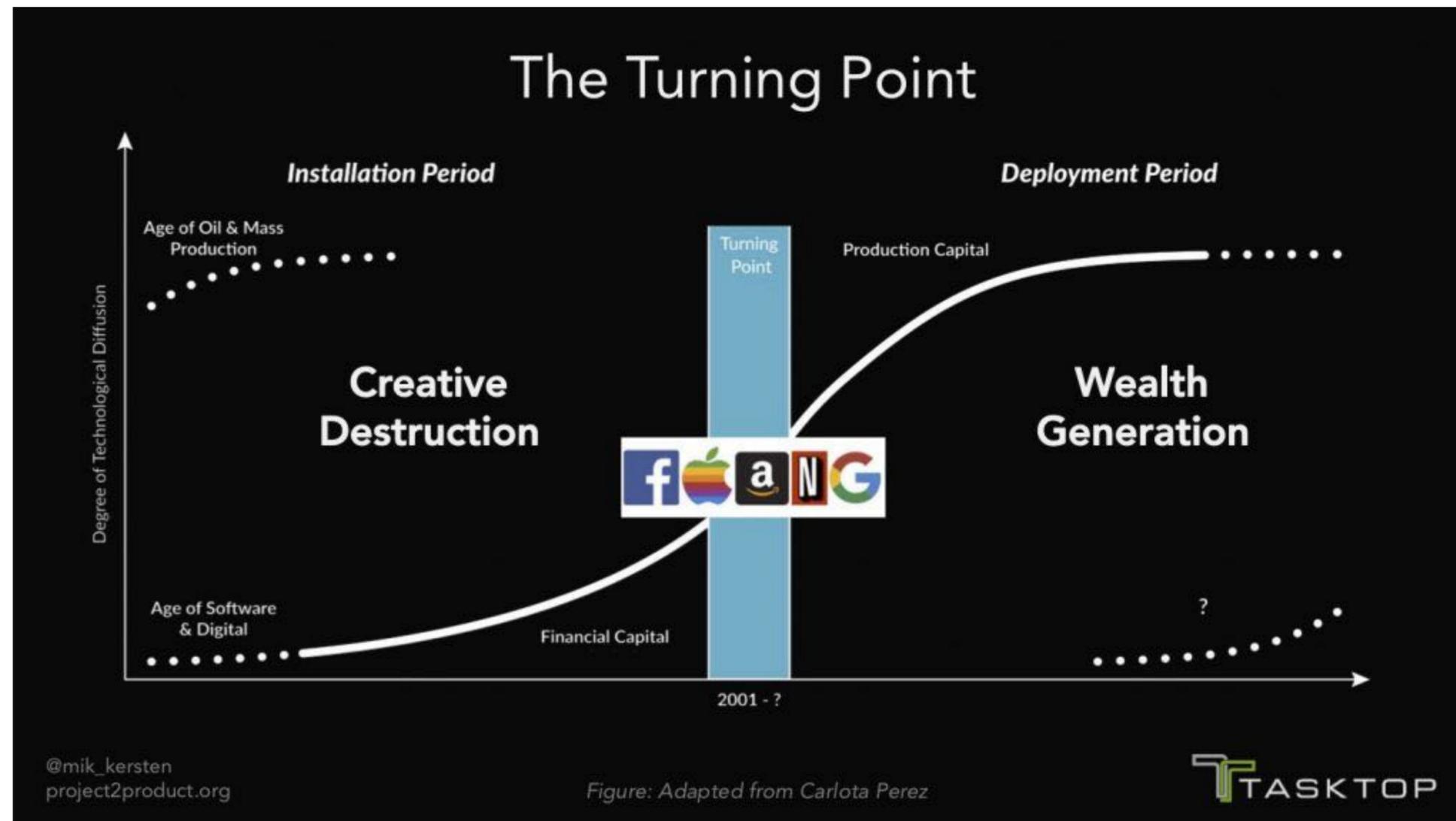
→ BRG Defect



Conclusion

Turning Point

- 지금이 DT와 AI 기술에 의한 터닝 포인트인가?



M. Kersten, "Five Predictions for the Coming Decades of Software," in IEEE Software, vol. 35, no. 5, 2018

Turning Point

- 터닝 포인트의 시간은 길지 않다

13 years from horse to car

Easter morning 1900: 5th Ave, New York City. Spot the automobile.



Source: US National Archives.

Easter morning 1913: 5th Ave, New York City. Spot the horse.



Source: George Grantham Bain Collection.

Thanks for Listening