

Module 14. 프로젝트 방법론

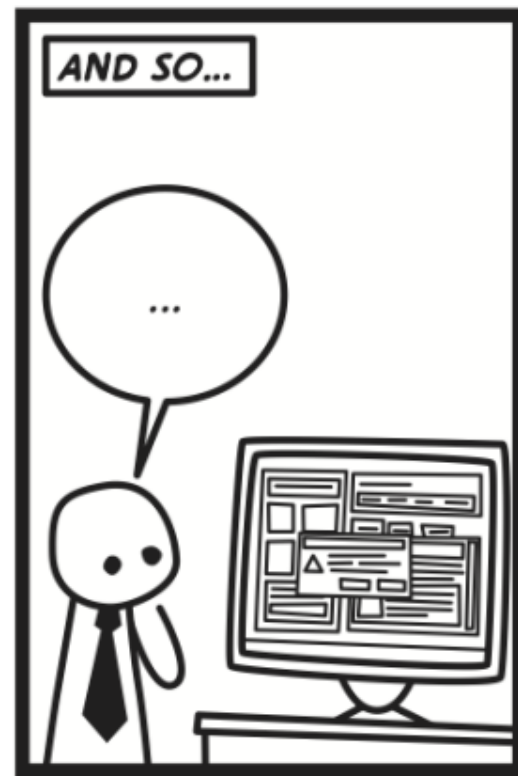
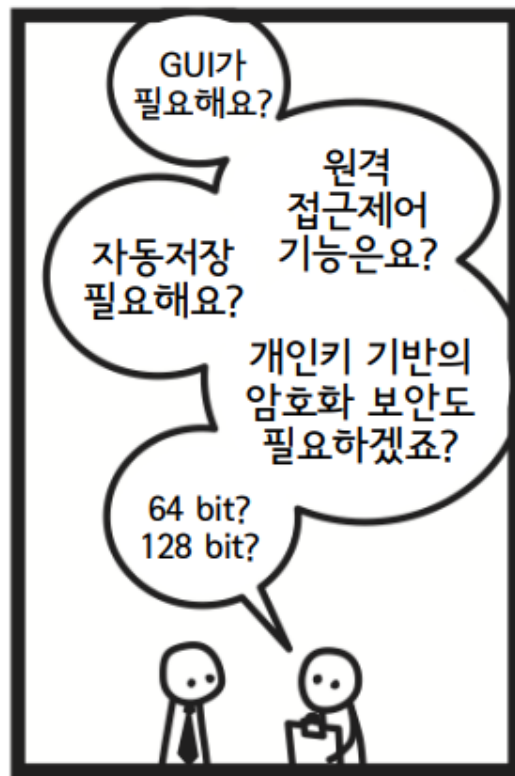
03

프로젝트 협업 도구

경북대학교 배준현 교수
(joonion@knu.ac.kr)

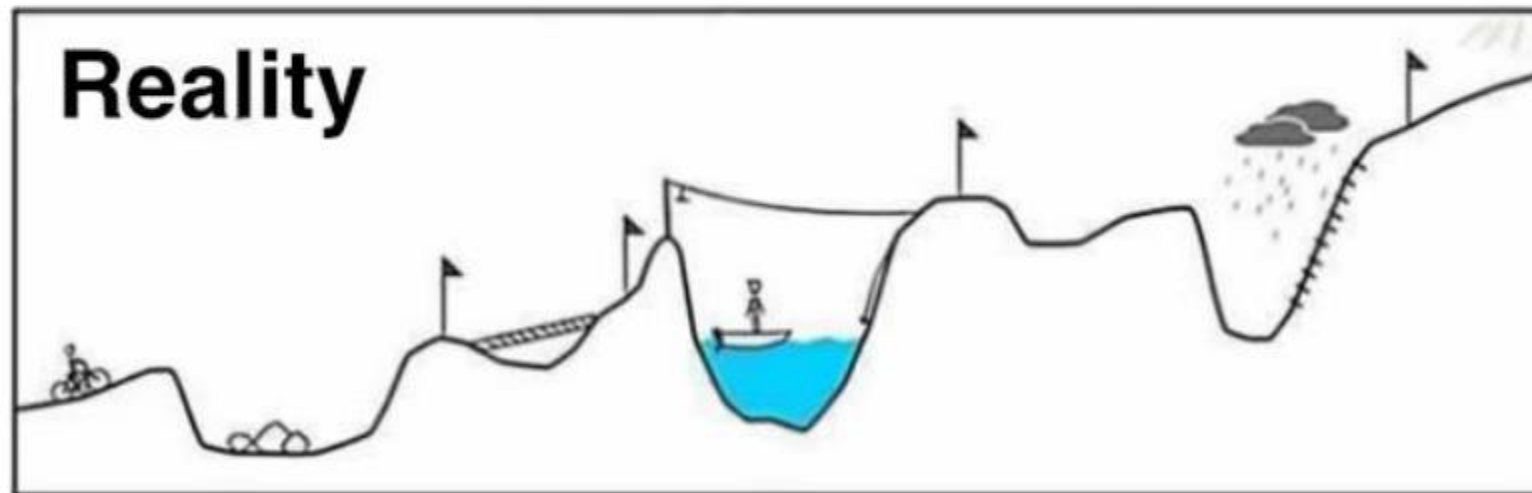
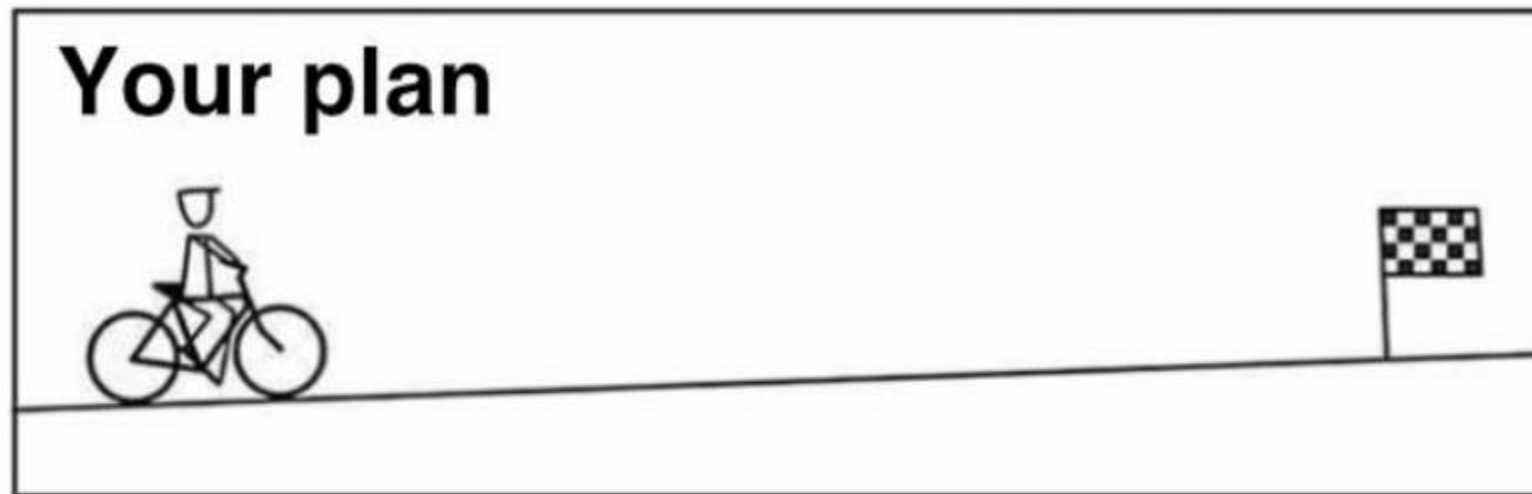


03. 프로젝트 협업 도구



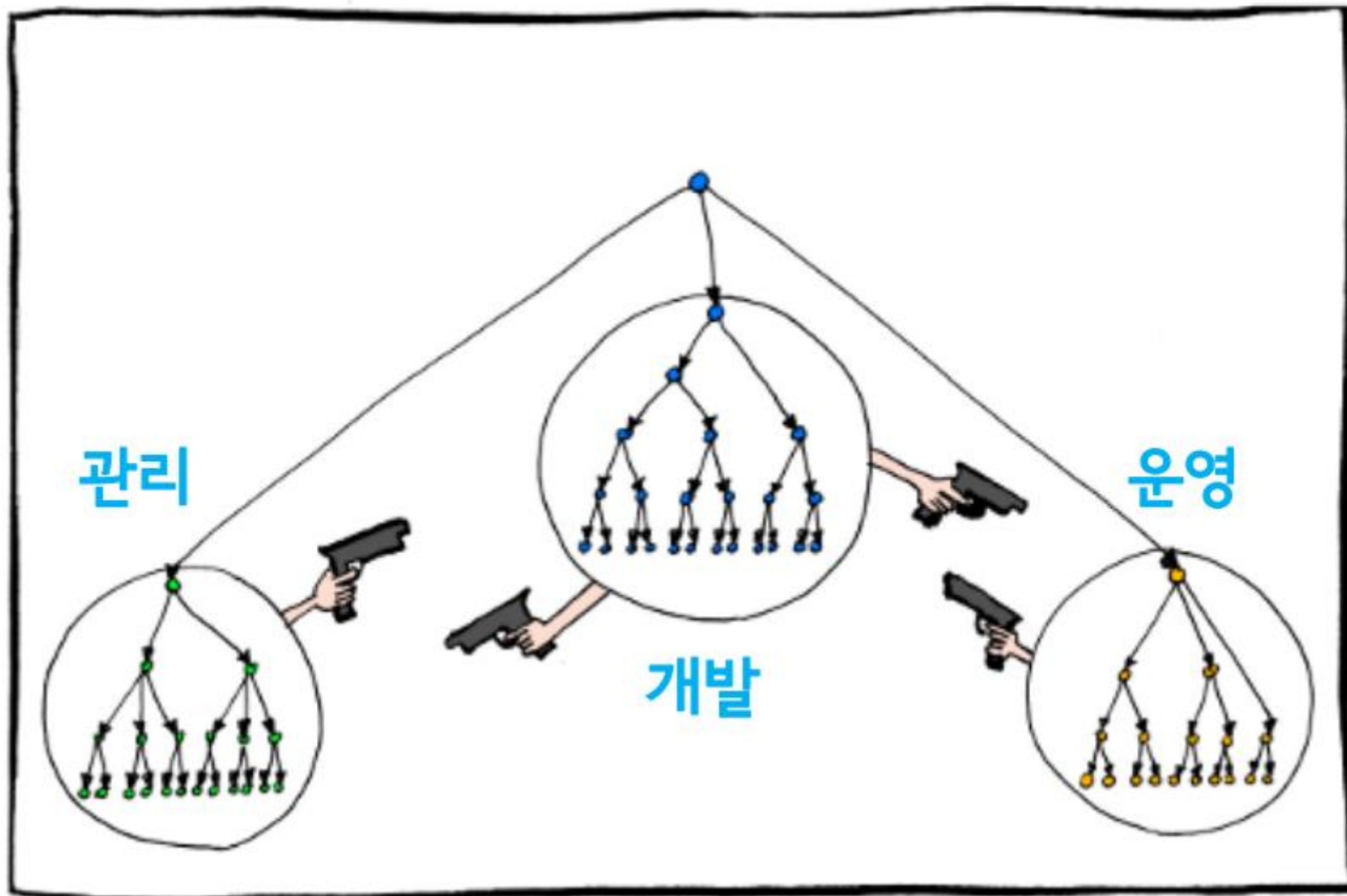


03. 프로젝트 협업 도구



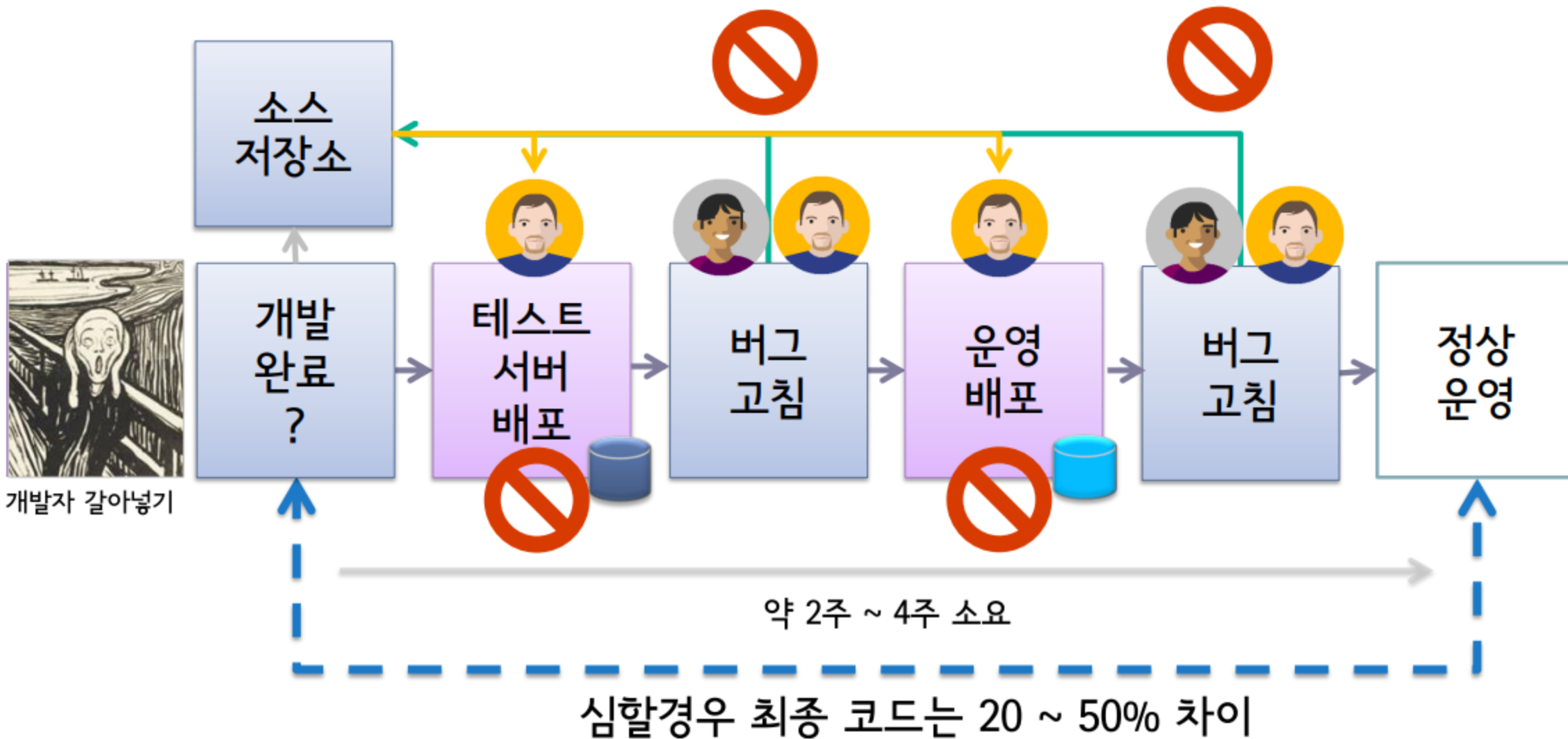


03. 프로젝트 협업 도구



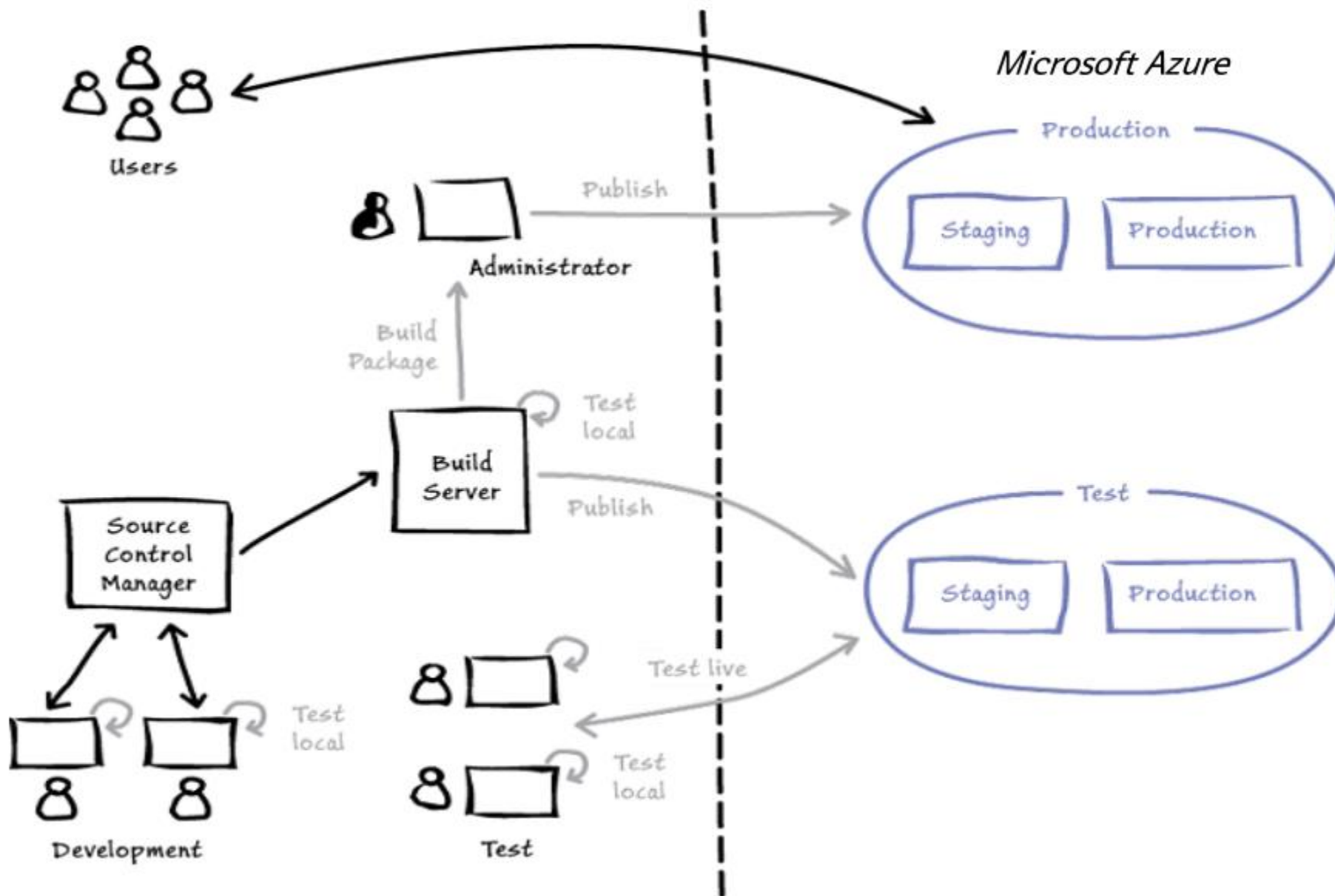


03. 프로젝트 협업 도구





03. 프로젝트 협업 도구





03. 프로젝트 협업 도구

N-tier

Loosely Coupled

Microservices

Serverless

Git

Agile/Scrum

2 Pizza team

CI/CT/CD

ALM

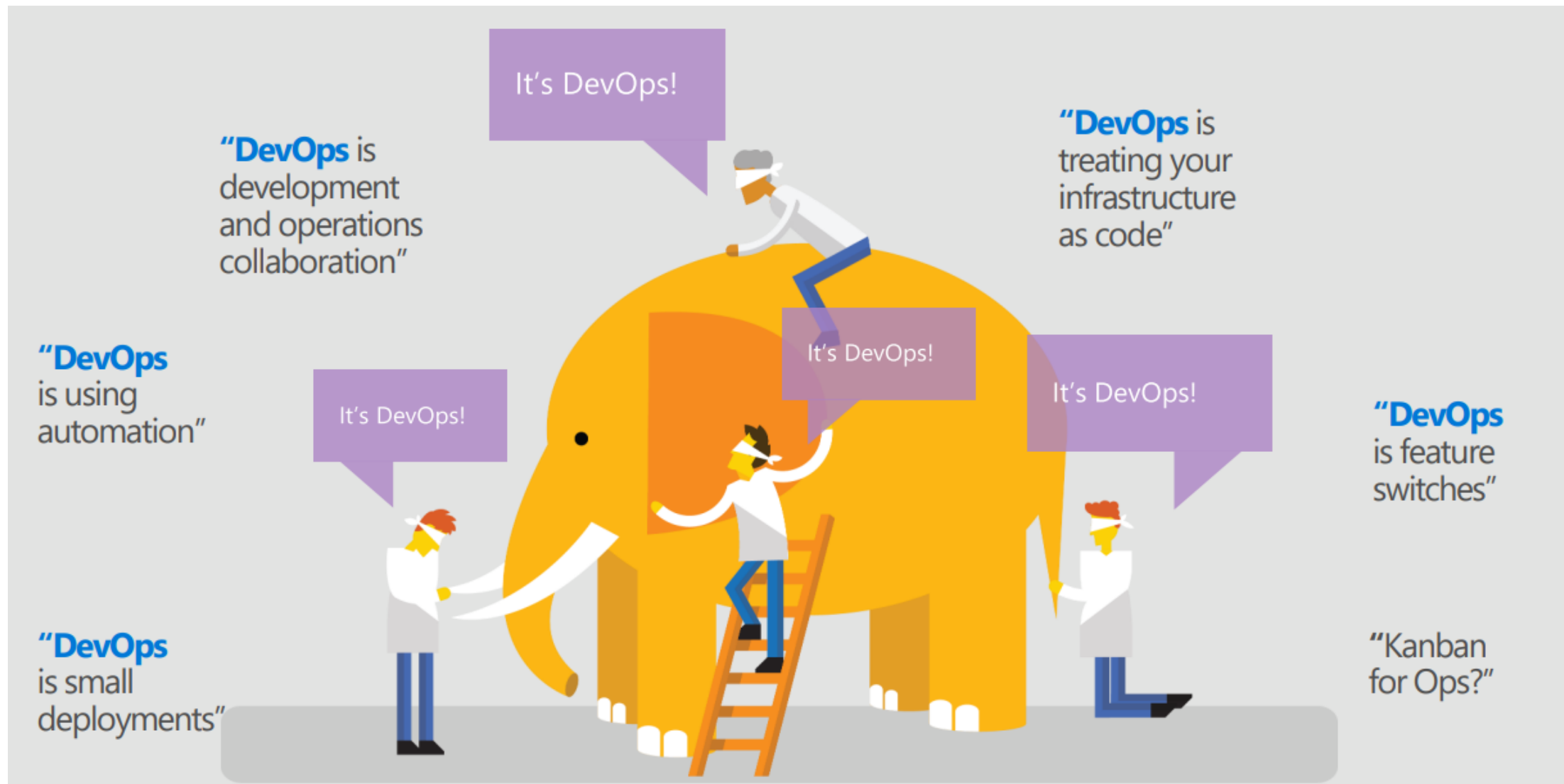
DevOps

DDD

TDD



03. 프로젝트 협업 도구





03. 프로젝트 협업 도구





03. 프로젝트 협업 도구

ALM



개발자 디자이너 PM

운영

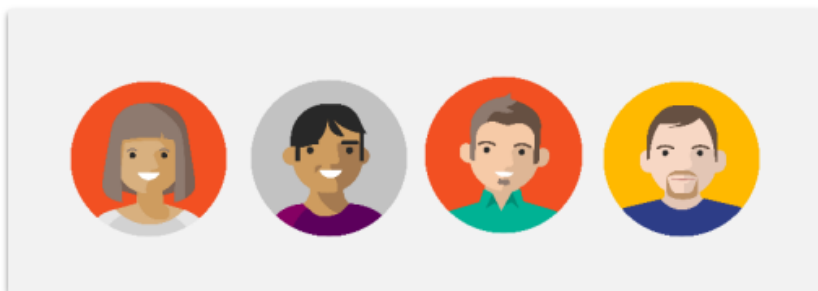


03. 프로젝트 협업 도구

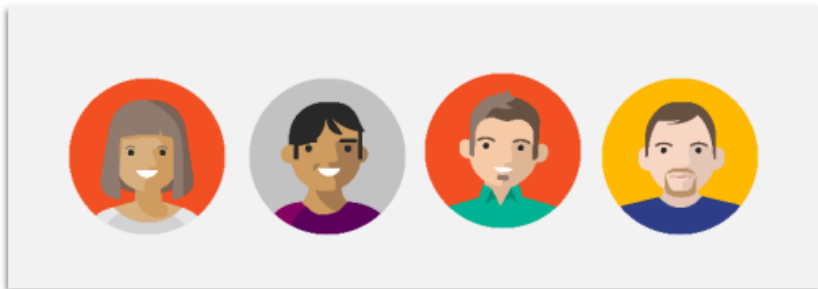
DevOps team

DevOps

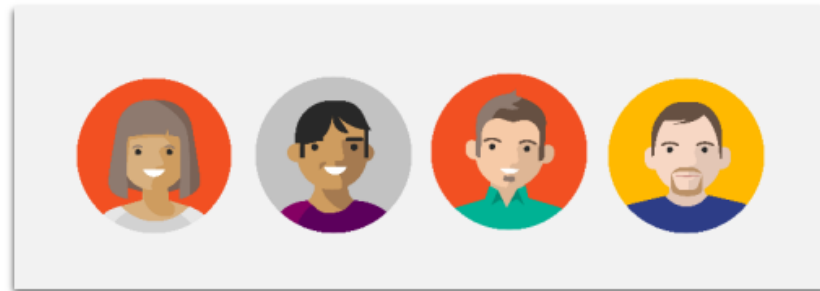
Feature Team (2 Pizza)



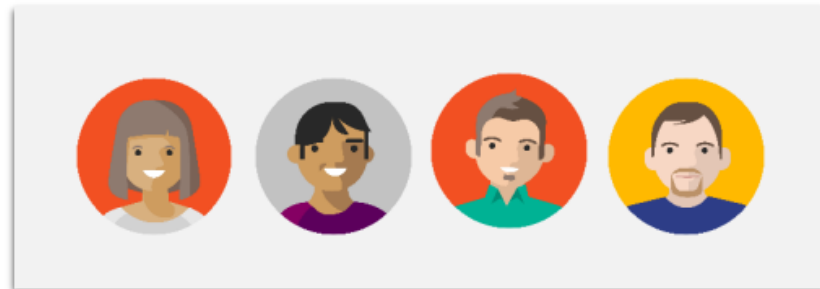
Feature Team (2 Pizza)



Feature Team (2 Pizza)



Feature Team (2 Pizza)



작은 규모의 팀이 신속하고 독립적으로 고객에게 가치를 제공할 수 있는 성과를 달성하는 것

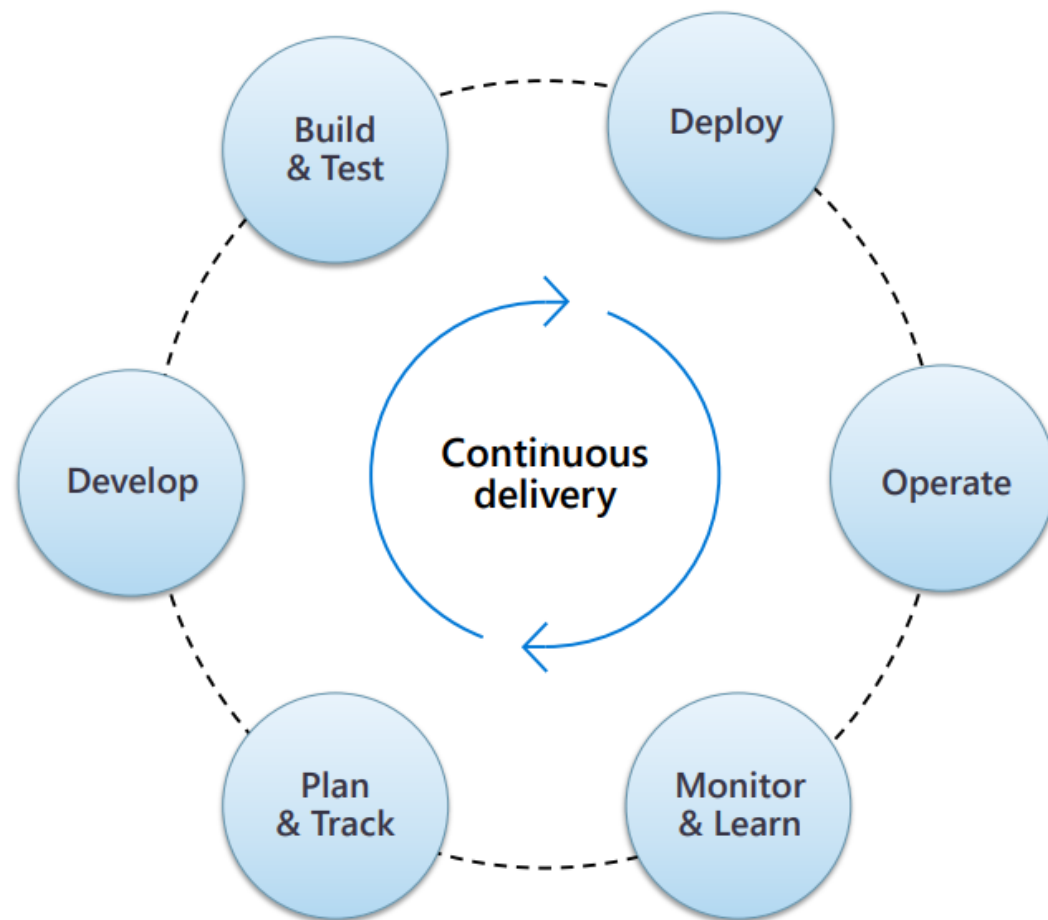


03. 프로젝트 협업 도구

DevOps 란

사람. 프로세스. 기술(도구)

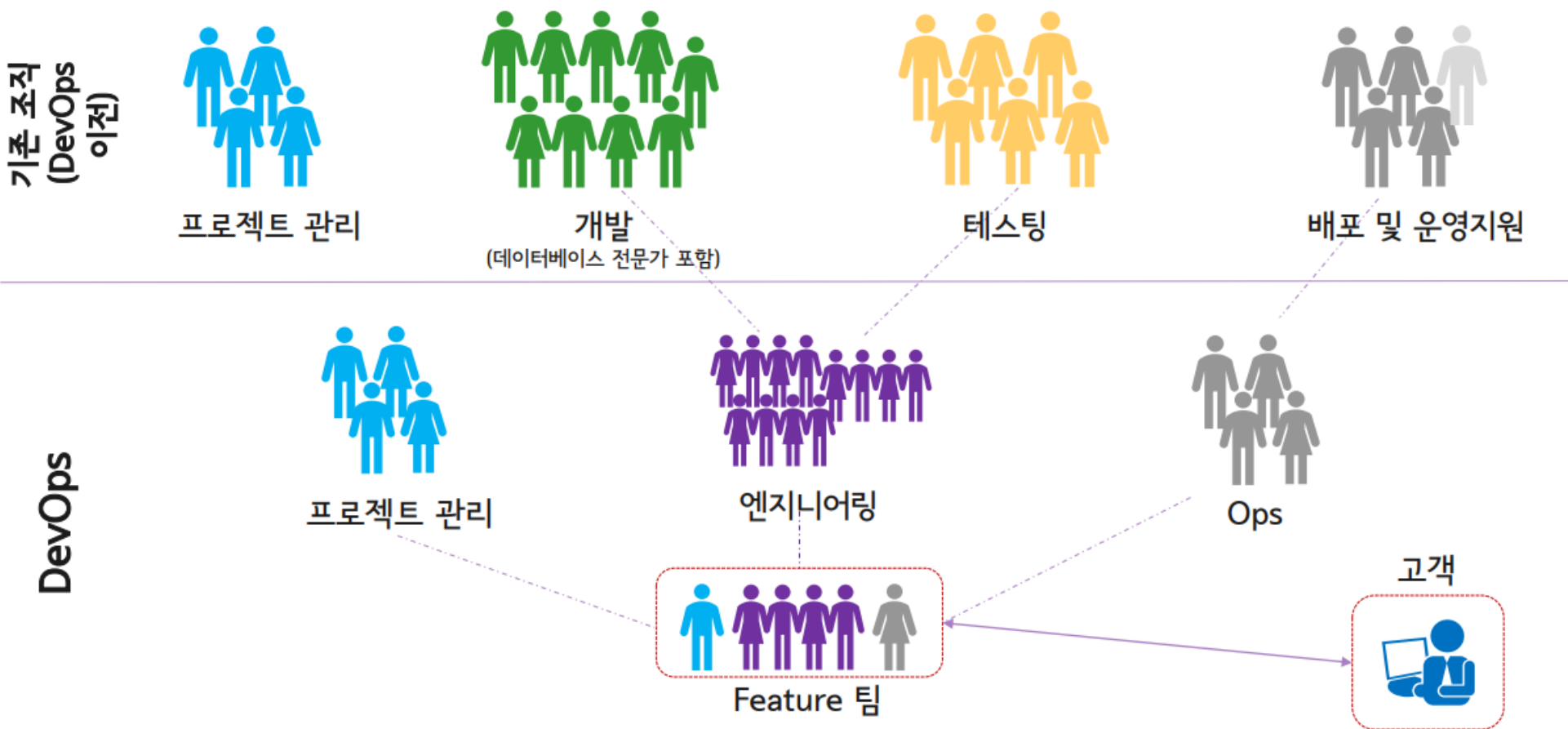
//
DevOps is the union of **people**,
process, and **technology** to
enable **continuous delivery** of
value to your end users.//





03. 프로젝트 협업 도구

DevOps를 위한 역할과 책임의 이동





03. 프로젝트 협업 도구

CI / CD

Continuous Integration

지속적인 통합(CI)은 개발 팀이 코드의 개발 및 테스트를 단순화하기 위해 사용하는 실행 방안이다. CI는 개발 초기에 버그나 문제를 파악하는 데 도움이 되며, 이를 통해 쉽고 빠르게 문제점을 해결할 수 있다. 자동화된 테스트와 빌드는 CI 프로세스의 일부로 수행된다. 프로세스는 정해진 일정에 따라 수행되거나 혹은 코드가 push될 때마다 실행된다. CI 시스템의 산출물로는 아티팩트가 생성된다. 아티팩트는 지속적인 딜리버리의 릴리스 파이프라인에서 사용되며, 이를 통해 배포 자동화를 구성할 수 있다.

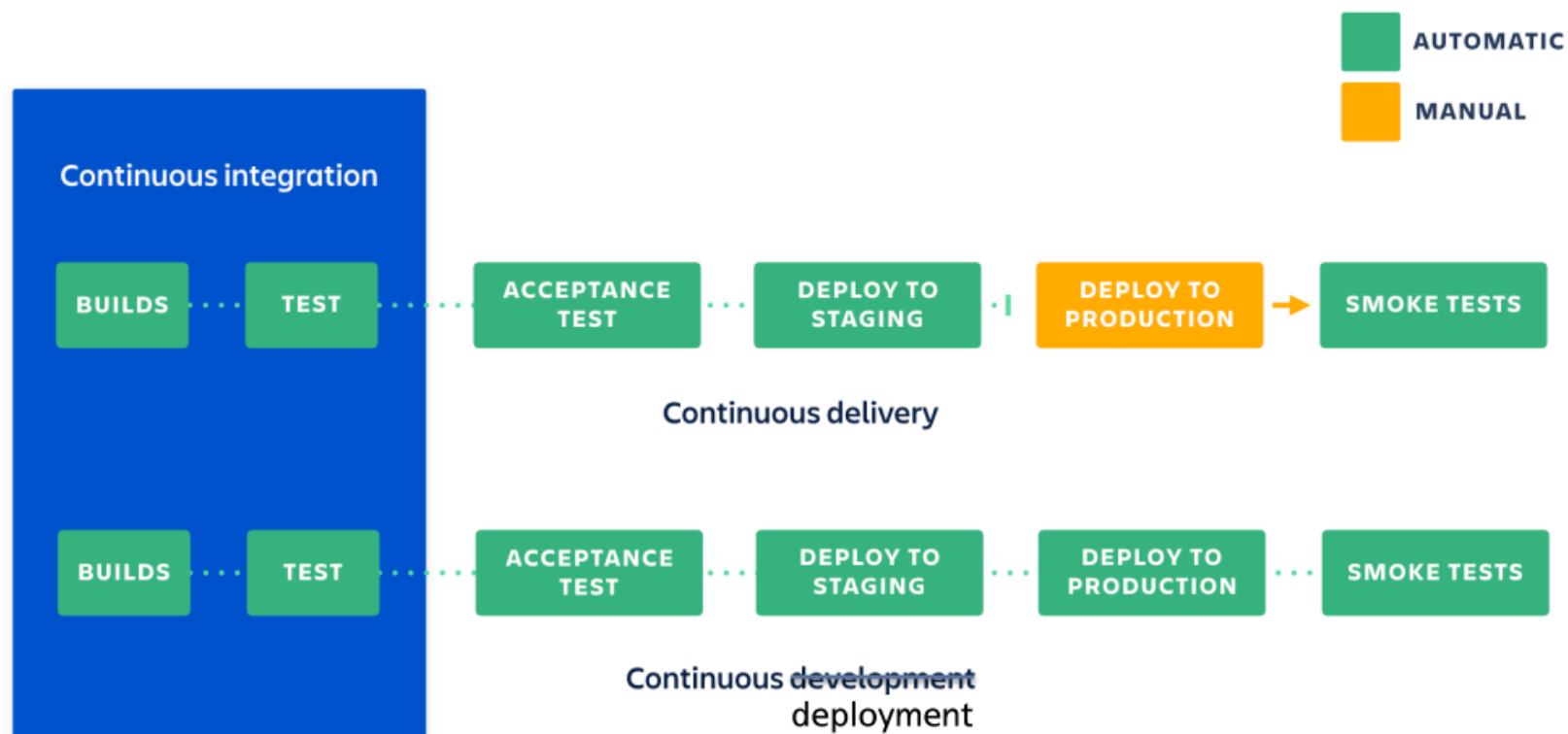
Continuous Delivery

지속적인 딜리버리(CD)는 코드를 하나 이상의 테스트 및 운영 환경에 빌드, 테스트 및 배포하는 프로세스다. 여러 단계로 배포 및 테스트를 실시하면 품질 향상시킬 수 있다. CI 시스템은 인프라와 애플리케이션을 포함하는, 배포 가능한 아티팩트를 생성한다. 자동화된 릴리즈 파이프라인은 이러한 아티팩트를 사용하여 기존 시스템에 새로운 버전과 수정 사항을 릴리즈한다. 모니터링 및 경고 시스템은 지속적으로 실행되어 전체 CD 프로세스에 대한 가시성을 확보한다. 이러한 시스템을 활용하면 에러를 빠르게, 자주 잡을 수 있다.



03. 프로젝트 협업 도구

CD(Delivery) 와 CD(Deployment)



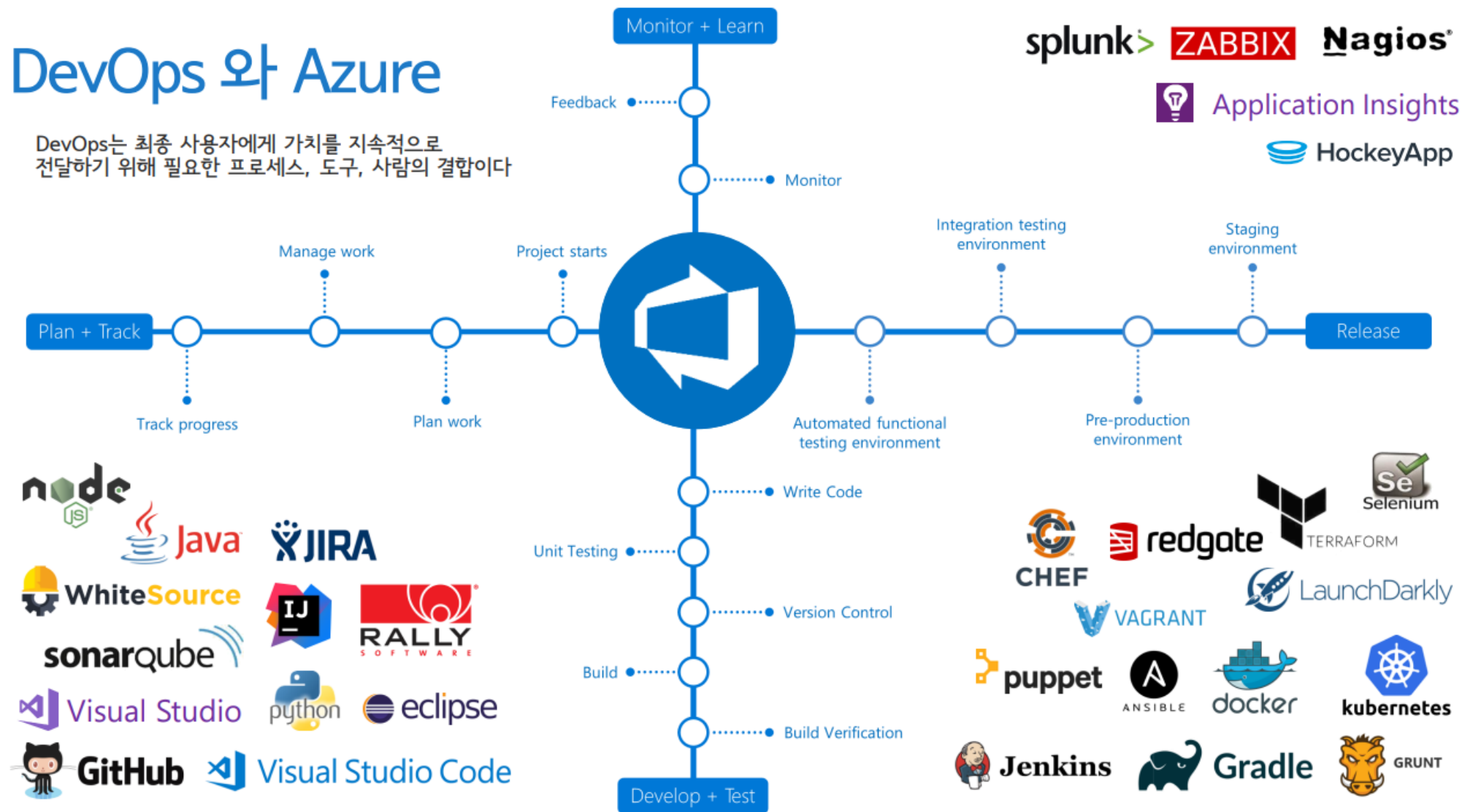
<https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>



03. 프로젝트 협업 도구

DevOps 와 Azure

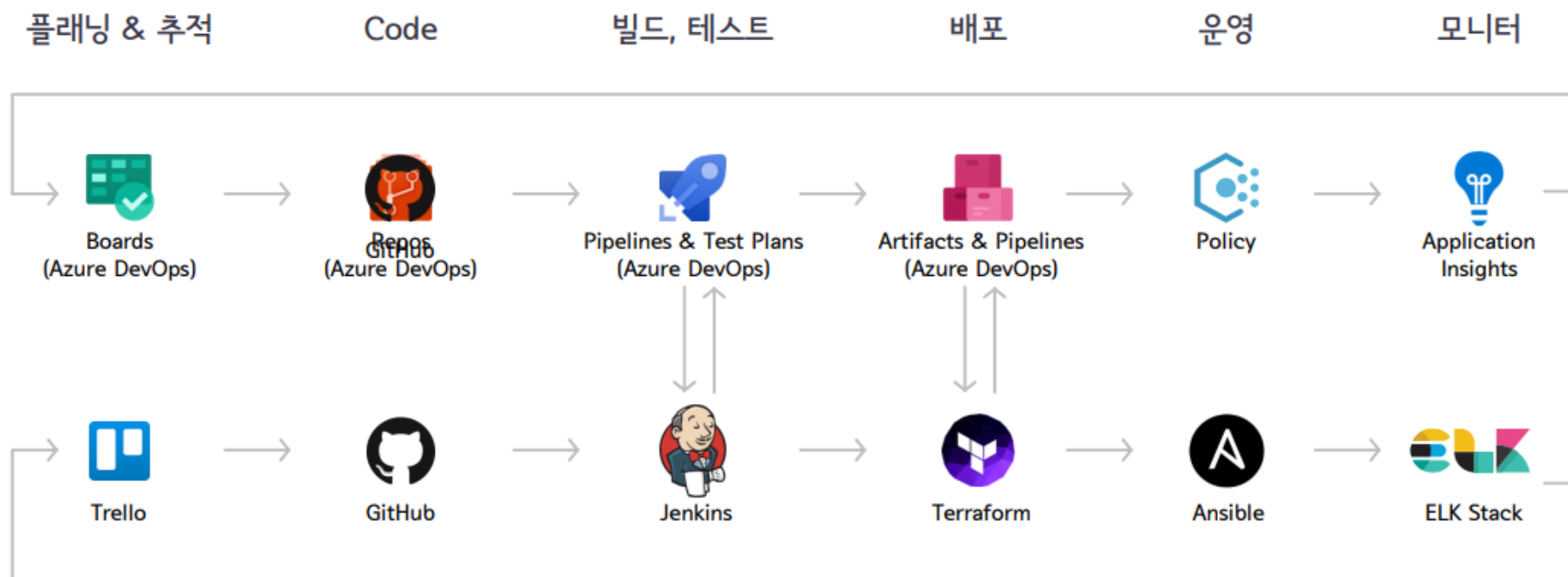
DevOps는 최종 사용자에게 가치를 지속적으로 전달하기 위해 필요한 프로세스, 도구, 사람의 결합이다





03. 프로젝트 협업 도구

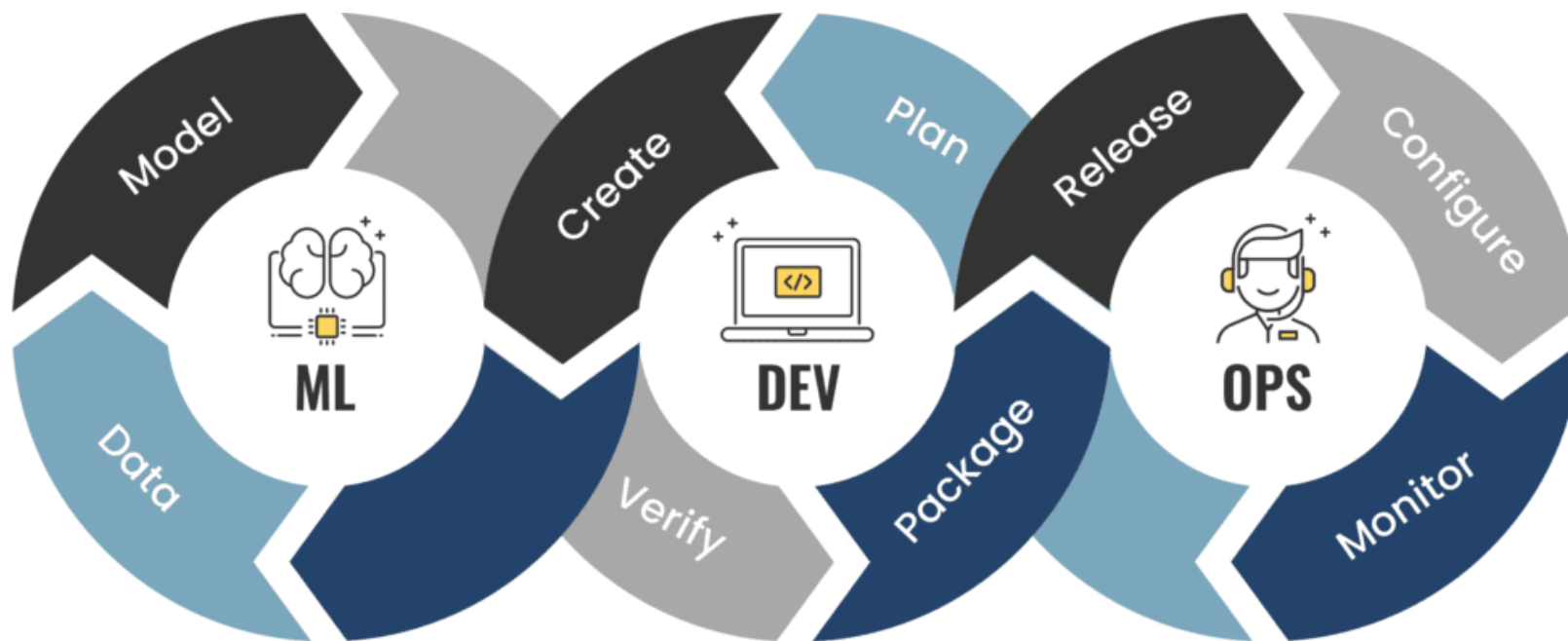
Azure 프레임워크 상에서의 DevOps





03. 프로젝트 협업 도구

- MLOps: Machine Learning Operations
 - MLOps = ML + Dev + Ops





03. 프로젝트 협업 도구

MLOps의 2개의 축



Data Scientists

- Quick iteration
- Frameworks they understand
- Best of breed tools
- No management headaches
- Unlimited scale



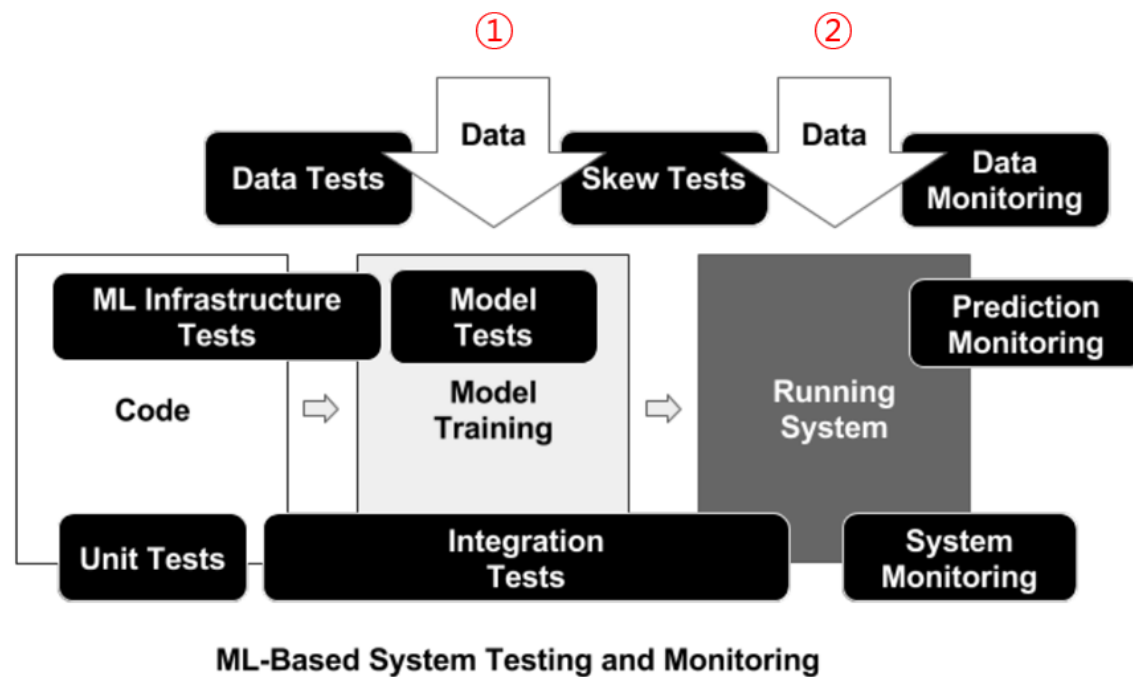
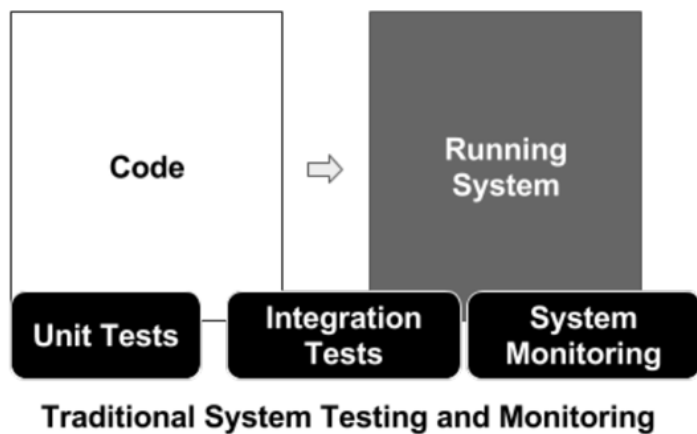
Data/Software Engineers

- Reuse of tooling and platforms
- Corporate compliance
- Observability
- Uptime



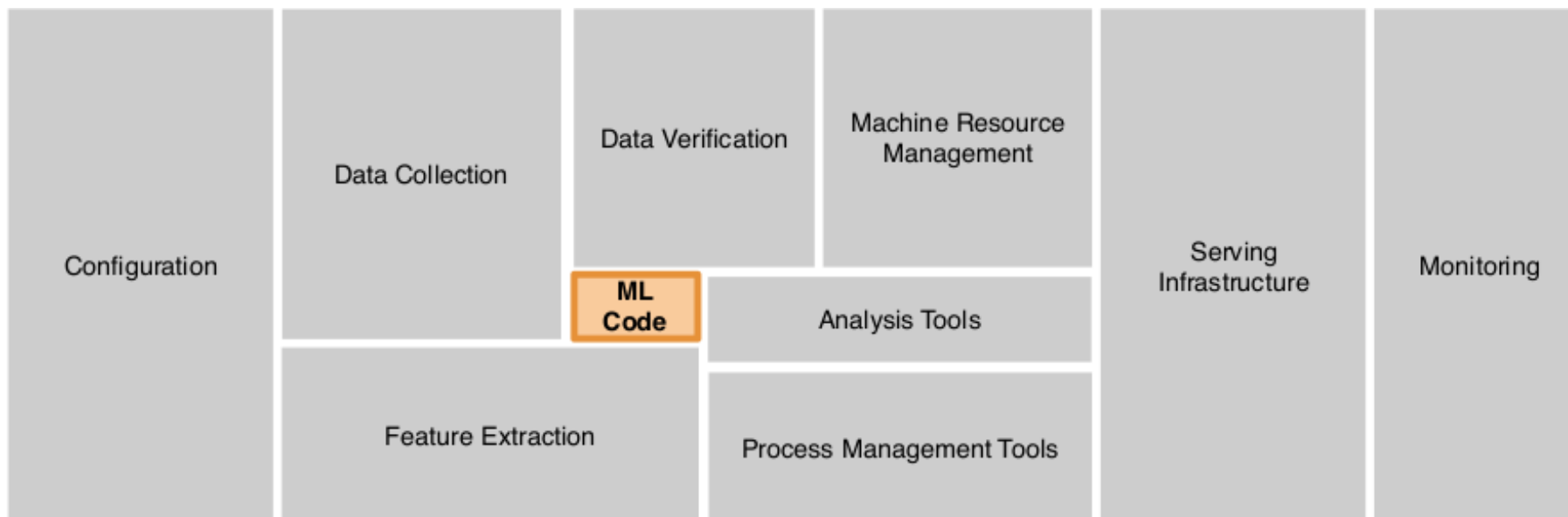
03. 프로젝트 협업 도구

ML 기반 시스템의 테스트와 모니터링



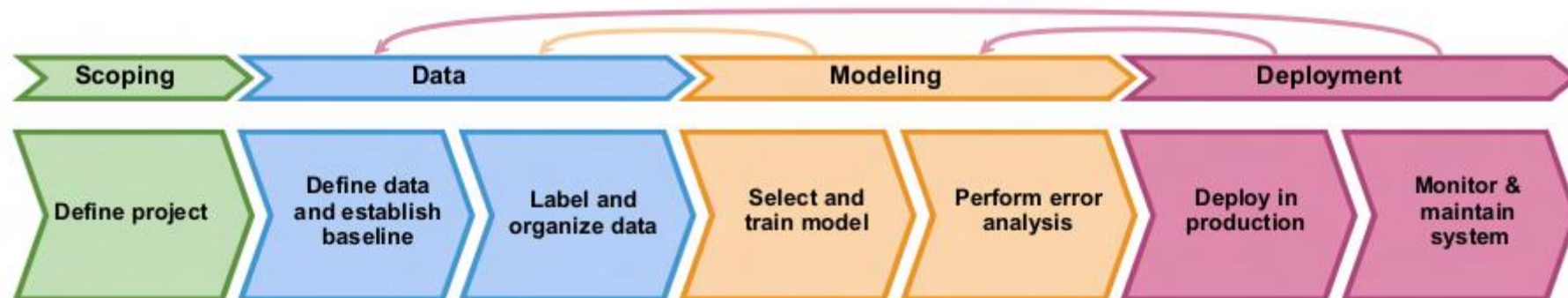


03. 프로젝트 협업 도구



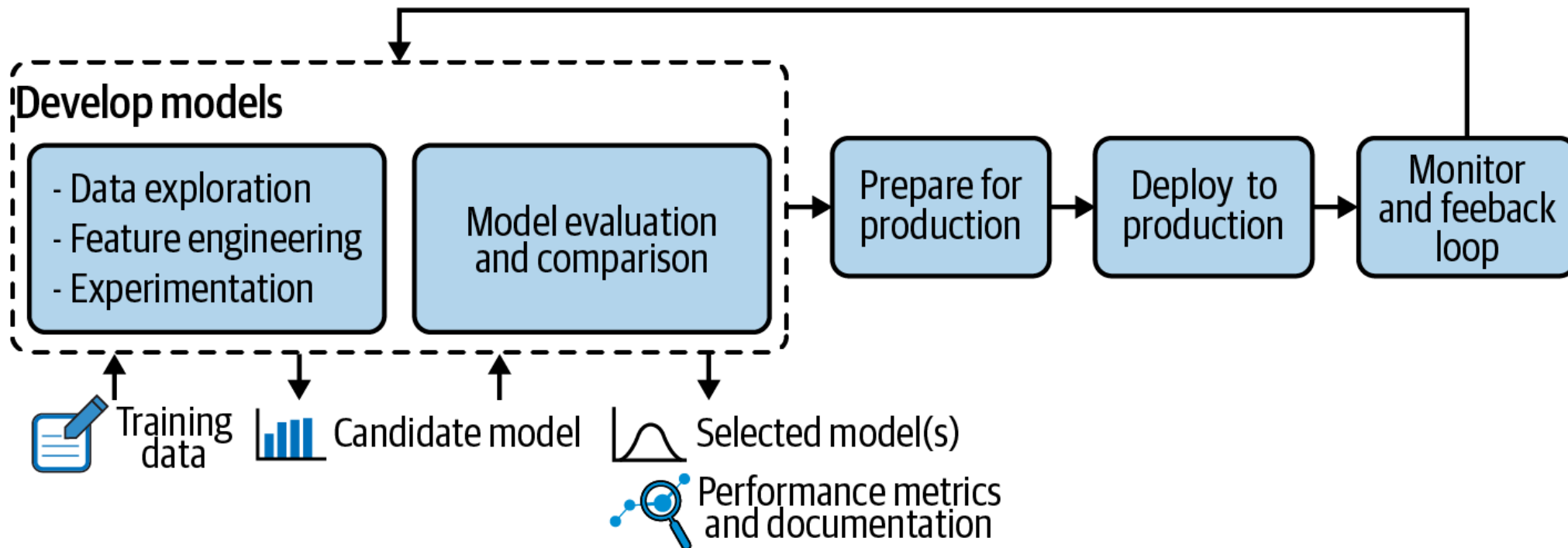


03. 프로젝트 협업 도구



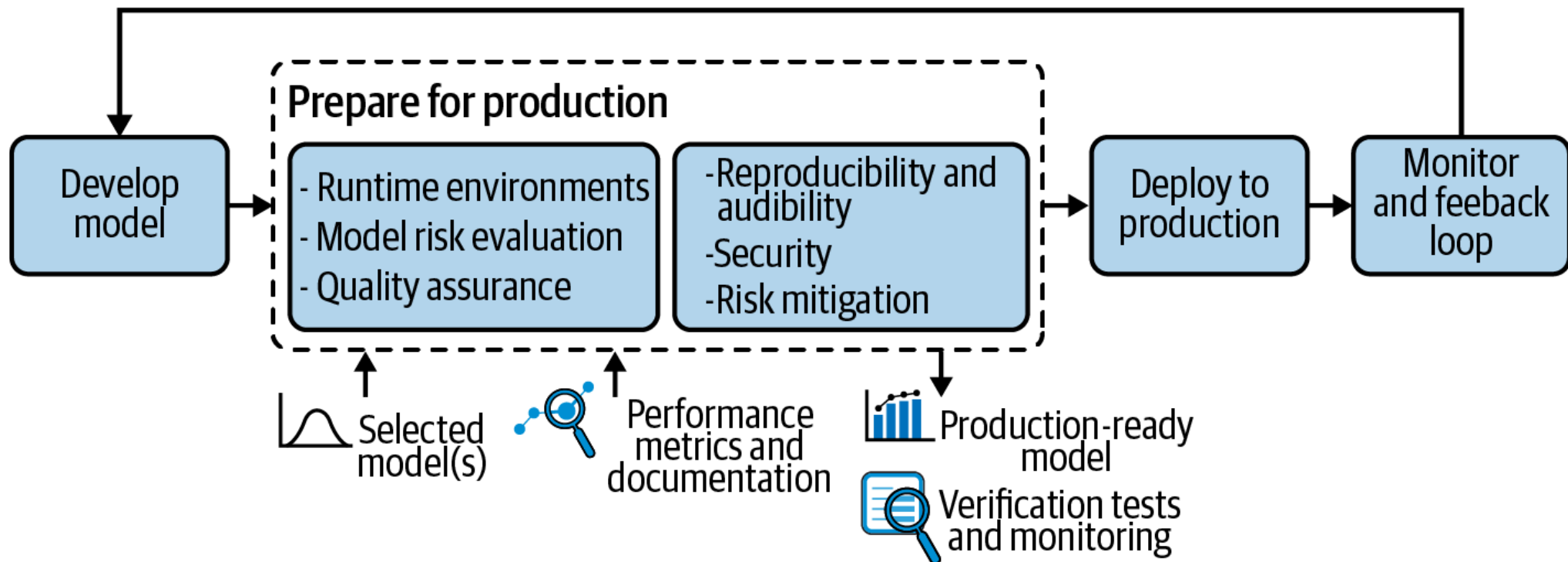


03. 프로젝트 협업 도구



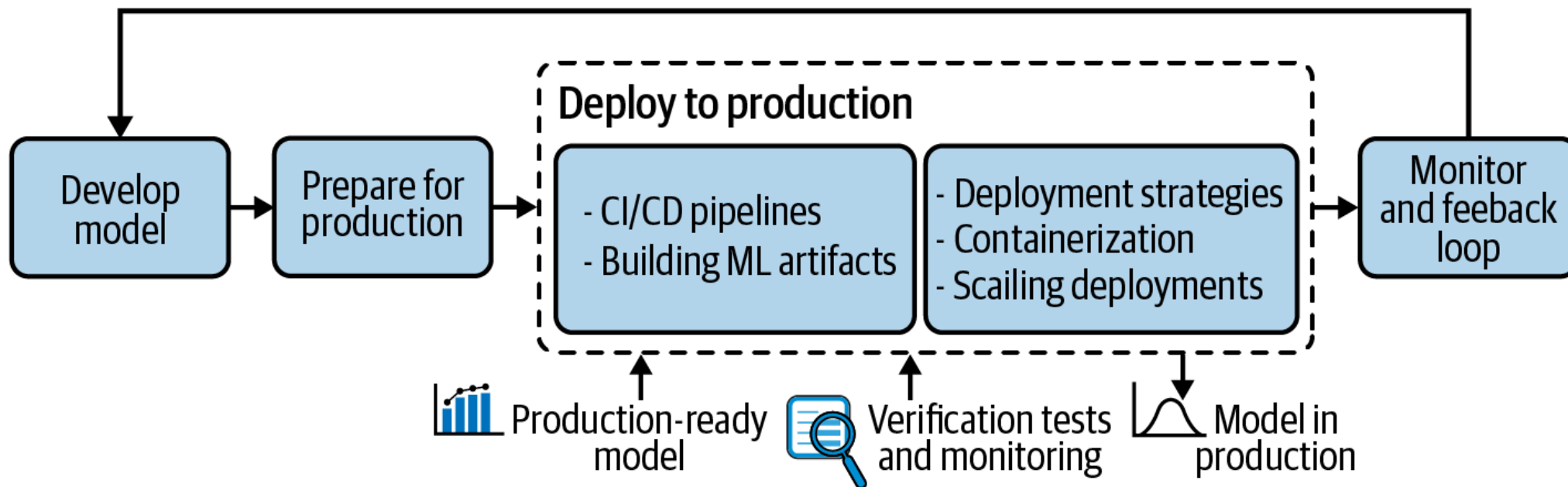


03. 프로젝트 협업 도구



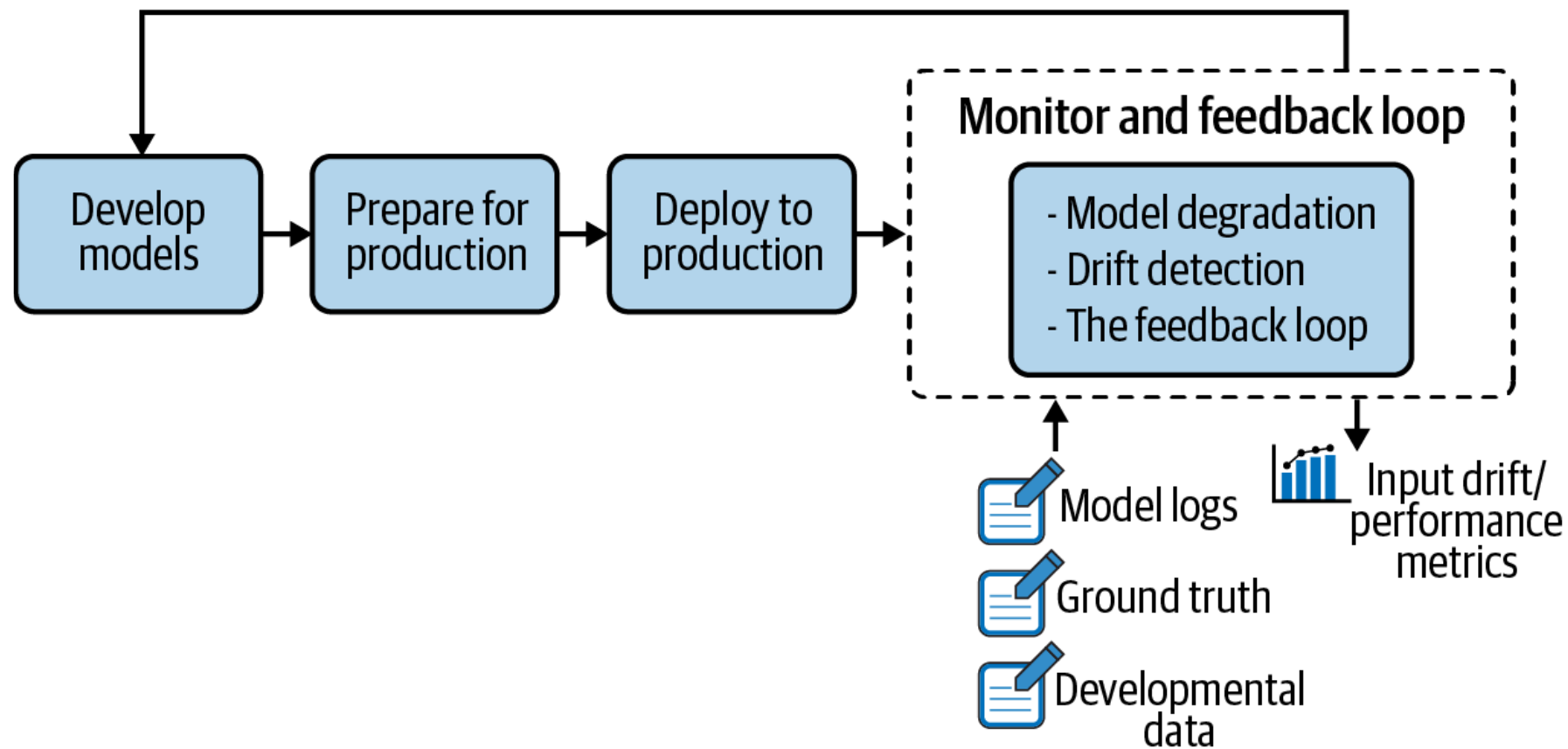


03. 프로젝트 협업 도구





03. 프로젝트 협업 도구



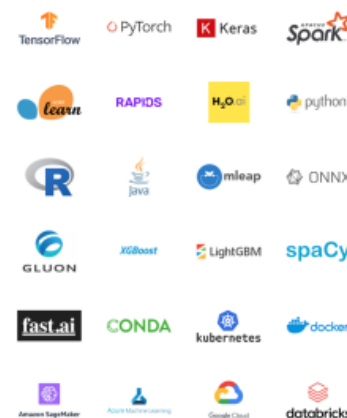
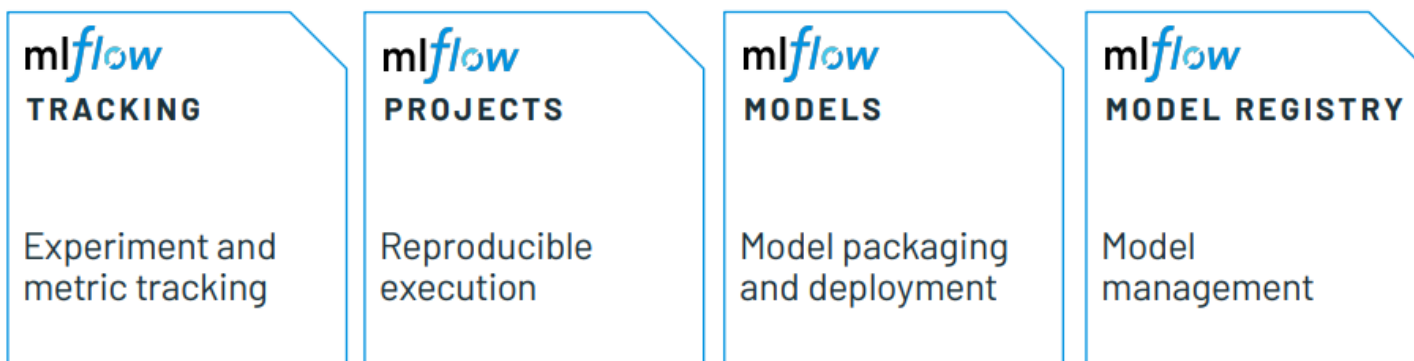


03. 프로젝트 협업 도구

mlflow: An Open Source ML Platform

Based on an **open interface** design philosophy: make it easy to connect arbitrary ML code & tools into the platform

- Simple command-line and REST APIs rather than environment-specific
- Easy to add to existing software



Built-in connectors

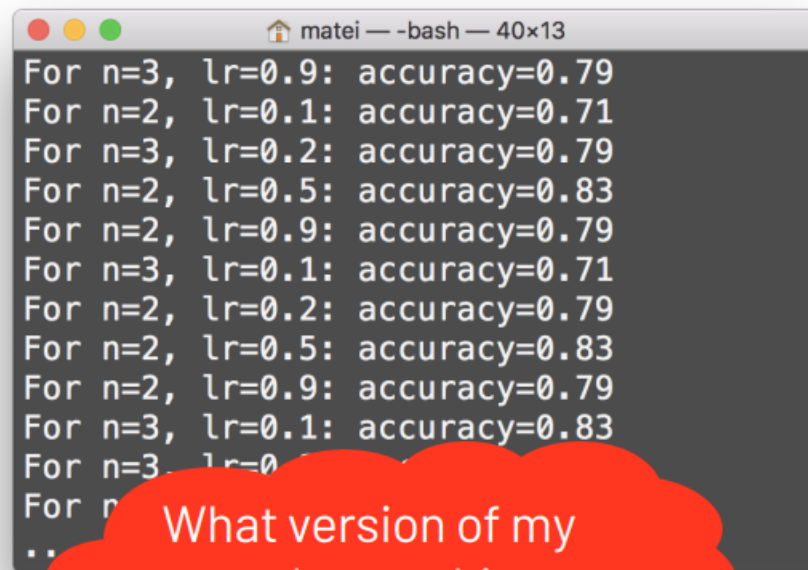


MLflow Tracking

Get visibility into experiments and production runs

```
data = load_text(file)
ngrams = extract_ngrams(data, N=n)
model = train_model(ngrams,
                    learning_rate=lr)
score = compute_accuracy(model)

print("For n=%d, lr=%f: accuracy=%f"
      % (n, lr, score))
```



What version of my code was this result from?





03. 프로젝트 협업 도구

MLflow Projects

Package code + dependencies for reusable workflows

my_project/
├── MLproject

├── conda.yaml
├── main.py
├── model.py
└── ...

```
conda_env: conda.yaml  
  
entry_points:  
  main:  
    parameters:  
      training_data: path  
      lr: {type: float, default: 0.1}  
    command: python main.py {training_data} {lr}
```

```
$ mlflow run git://<my_project>
```

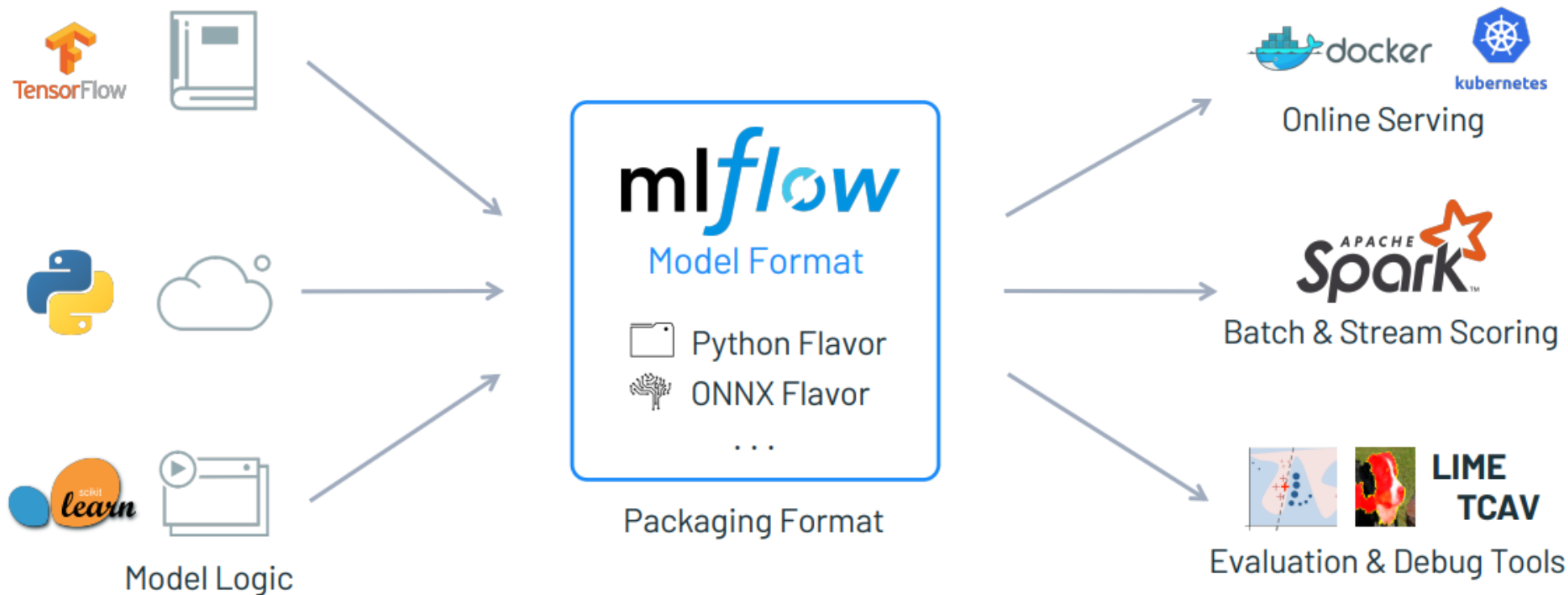
```
mlflow.run("git://<my_project>", ...)
```



03. 프로젝트 협업 도구

MLflow Models

Generic format to package & deploy models from any library

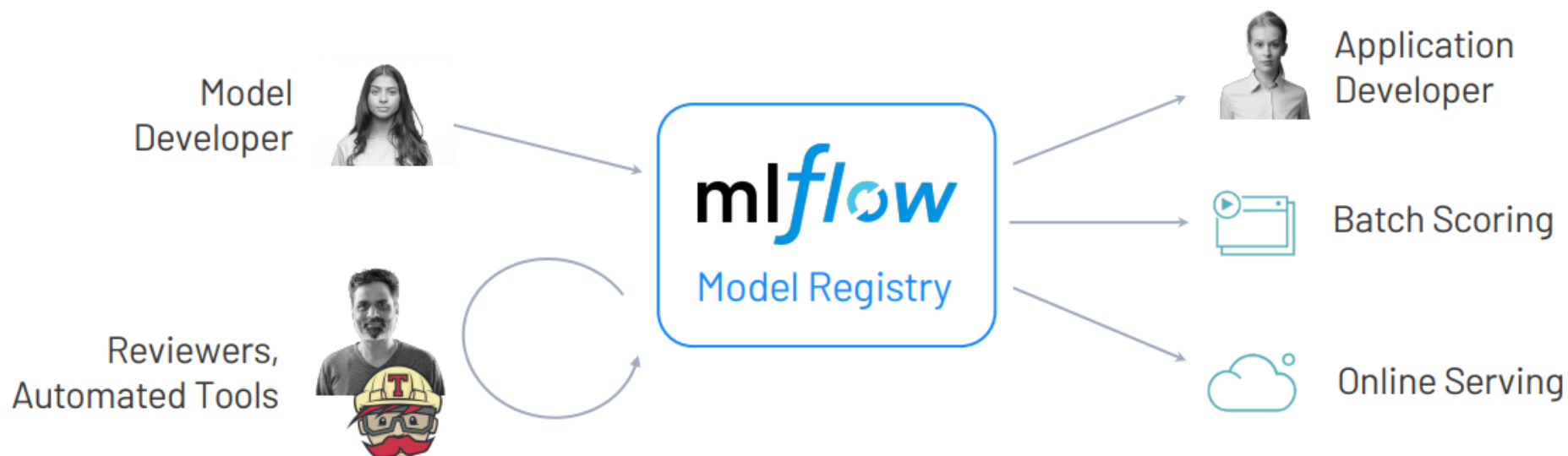




03. 프로젝트 협업 도구

MLflow Model Registry

GitHub-like environment for managing and reviewing models





03. 프로젝트 협업 도구

Tag and Search APIs for Automated CI/CD

Tags to track custom metadata for a model version, e.g. test results

Search API to automate model management and MLOps actions



mlflow Experiments Models

Models > KNN > Version 12 ▾

Created at: 2018-12-04 17:11:06 User: test@example.com Stage: Staging ▾

Last Modified: 2018-12-04 17:11:06 Source: Run 123

▸ Notes

▸ Pending Requests

▾ Tags

Name	Value	Actions
passed-gdpr-review	true	
passed-performance-test	true	

[Add Tag](#)

▾ Activity

Alice requested a stage transition None → Staging . 10 hours ago

Carol approved a stage transition None → Staging . 10 hours ago

Any Questions?

