

WEEK 3 : Design

Hosting & Staging Setup
Finalize API Goals

Infrastructure Recs:
- Laravel + Backbone

1) WEEK 4 : Design

Initialize Database

Normalize Data

Create Fixtures
(Testing)

IT Project Overview

2) WEEK 5 : Design

Admin Dashboard
B/E Frame work

B/E Sorting
(for All
Dashboard
Views)
w/ Pagination

B/E Filter
(for all views)

B/E Cmd for
all DB
Tables

Exporting

Database
API
(continuous)

Engineering

3) WEEK 6 : Dev

ADMIN

Software Engineering

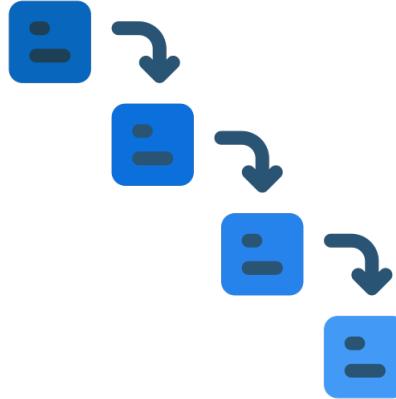
API

4) WEEK 7 : Dev

Parash
suggest

프로젝트 개발 방법론

프로젝트 관리를 하기 위해서 다양한 방법론이 존재하는데, 그 중 가장 대표적인 방법론은 Waterfall과 Agile이라고 할 수 있습니다.



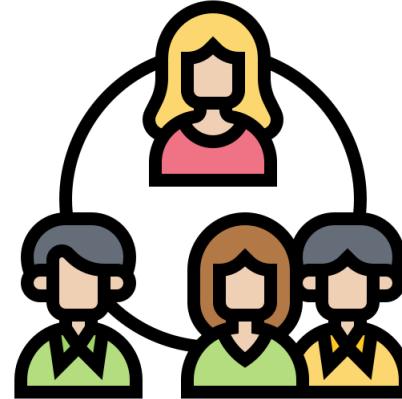
Waterfall

- 가장 대표적인 프로젝트 관리 방법론
- 선형 접근 방법
- 요구사항 분석에서 테스트에 이르는 전체 과정을 단계적으로 수행
- 단순하고 직관적
- 유연성 부족하고 변화에 대처하기 어려움



Agile

- Waterfall의 단점을 보완하고자 나온 방법론
- 반복적, 점진적 개발을 지향
- 방법론이라기 보다는 프로젝트 관리 철학에 가까움
- XP, Scrum, Lean, Kanban 등이 있음



Scrum

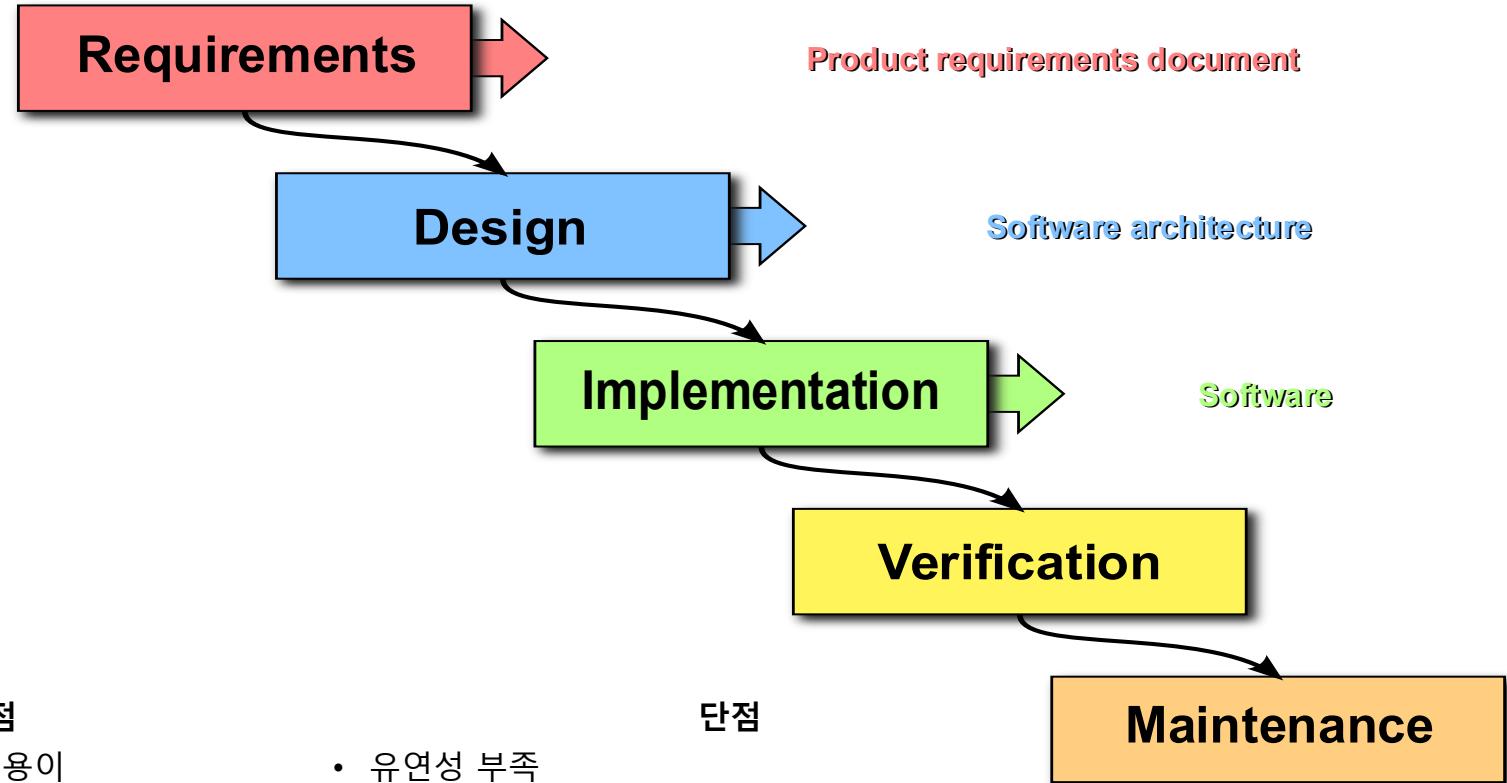
- Agile 방법론 중 가장 대표적인 방법
- 럭비의 스크럼 대형처럼 팀이 단합해서 목표를 향해 간다는 의미
- 작업을 2-4주 sprint로 나눠서 진행



Lean

- Toyota Production System에서 유래
- 생산에서 물리적인 낭비를 줄인다면 소프트웨어 개발에서는 낭비 요소적인 관행을 제거
- 낭비, 불균형, 무리를 없애는 방향으로 진행

Waterfall 모델



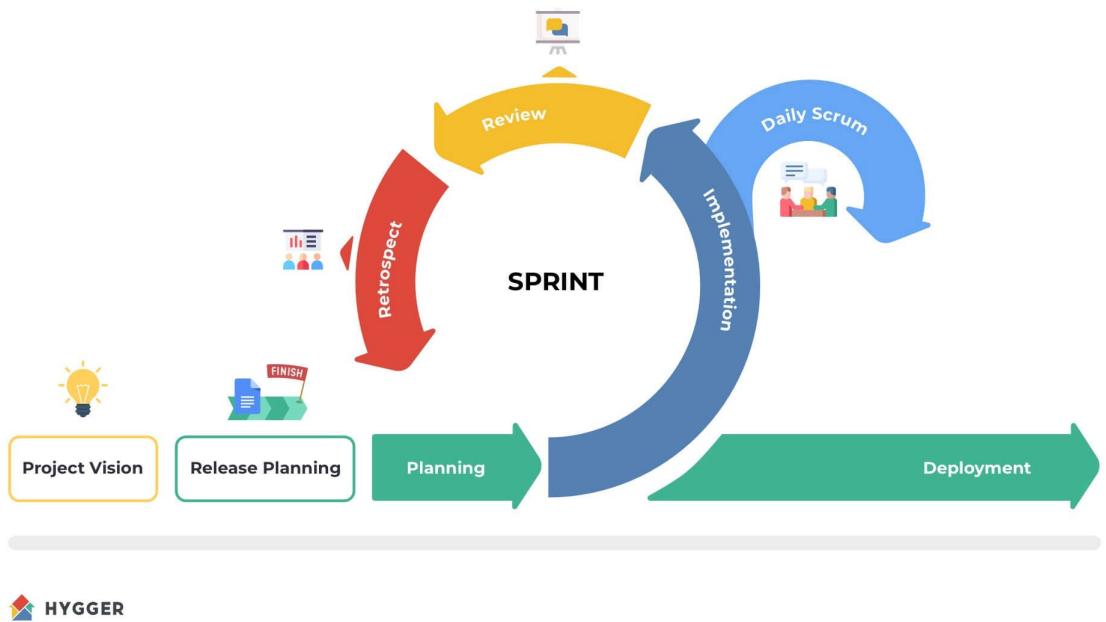
장점

- 전체 과정에 대한 이해가 용이
- 현재 단계에 대한 이해가 용이
- 고전적인 방법론으로 적용 사례가 풍부
- 문서, 산출물의 관리, 적용 용이
- 통제, 위임하기 용이

단점

- 유연성 부족
- 단계별 진척도 추적 어려움
- 순서 변경 등 어려움
- 테스트 단계에 발견된 중요 결함에 대한 대응 어려움
- 고객 요구사항에 대한 상세한 반영 어려움

Scrum 모델



장점

- 빠른 결과를 얻을 수 있음
- 동기 부여, 생산성 높음
- 긴 프로세스를 작은 단위로 나눠서 관리
- 프로젝트 가시성 높음

단점

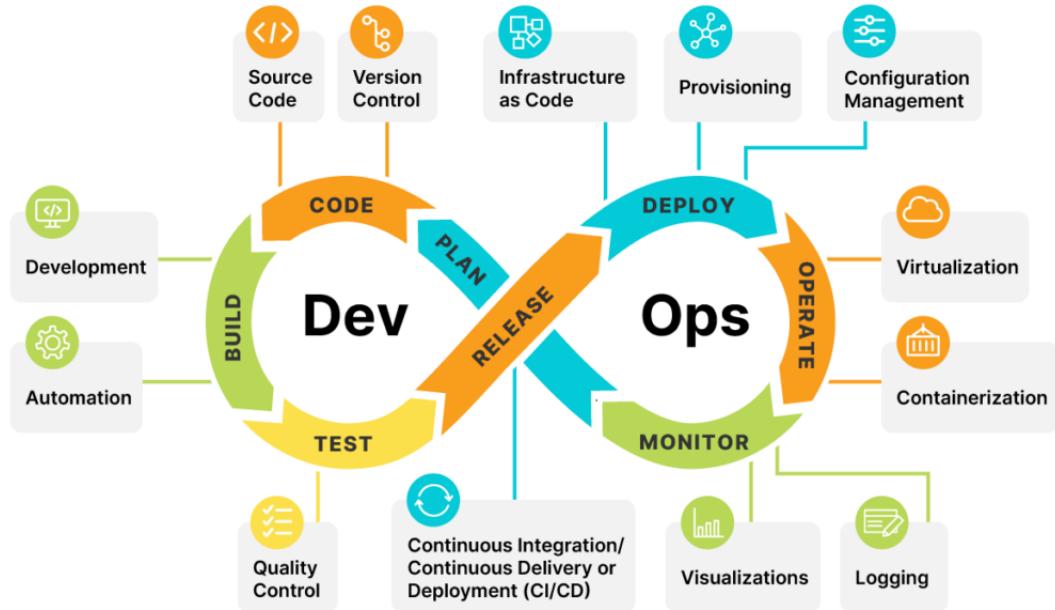
- 경험 있는 인력 필요
- 팀의 자율성 필요
- 대규모에 적용하기에 어려울 수 있음



<https://hygger.io/blog/what-is-scrum-lifecycle/>

DevOps (데브옵스)

소프트웨어 개발(development)과 운영(operation)의 약자 Dev와 Ops를 합친 말로, 개발 따로, 운영 따로 하던 관행을 벗어나 보다 더 효율적이고 안정적인 소프트웨어를 제공하기 위한 방법론입니다.



<https://productcoalition.com/12-top-devops-best-practices-for-a-successful-transition-in-2023-b73b54014d0d>

Ask ChatGPT

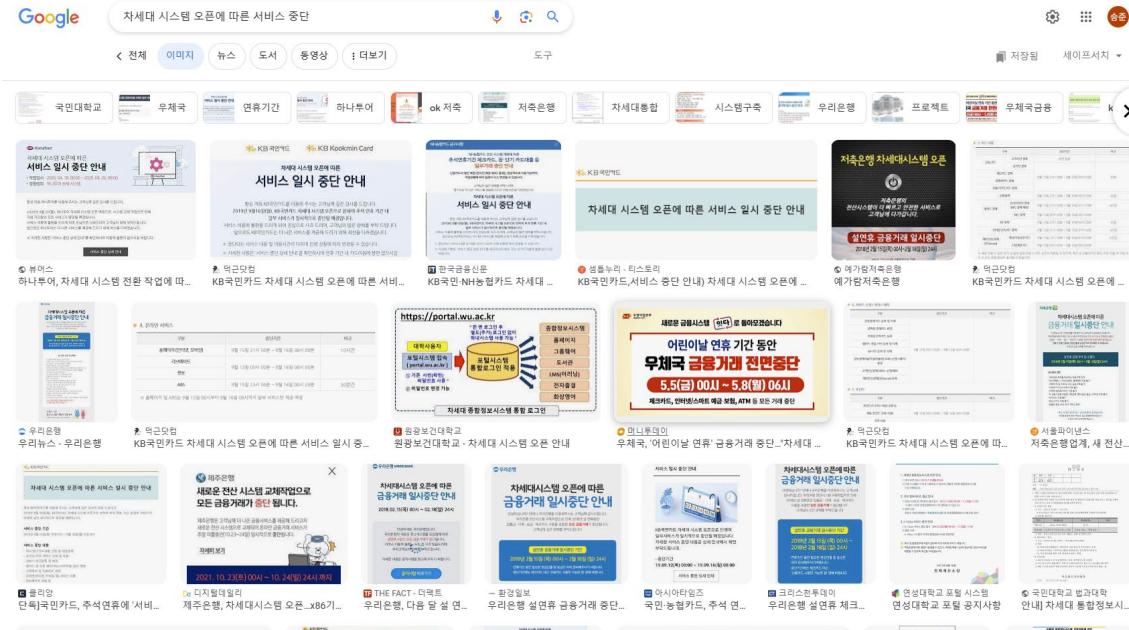
데브옵스란?

데브옵스(DevOps)는 **개발(Dev)과 운영(Ops) 팀 간의 협업, 커뮤니케이션, 통합을 개선하는 일련의 관행, 방법론 및 문화 철학**입니다. 데브옵스의 주요 목표는 더 효율적이고 안정적인 소프트웨어 제공을 달성하면서 지속적인 개선과 혁신을 장려하는 것입니다. 데브옵스의 주요 특징은 다음과 같습니다:

- 지속적 통합 (CI):
- 지속적 제공 (CD):
- 인프라스트럭처 코드 (IaC):
- 자동화된 배포:
- 모니터링 및 피드백 루프:
- 협업과 커뮤니케이션:
- 공유 책임:

Continuous Integration & Continuous Delivery (지속적인 통합 및 배포)

CI/CD는 현대 소프트웨어 개발 방법론의 핵심 아이디어 중 하나로서 Digital Transformation (디지털 전환)이 핵심 비즈니스 역량이 되는 시대에 기업이 반드시 갖춰야 할 요소입니다.



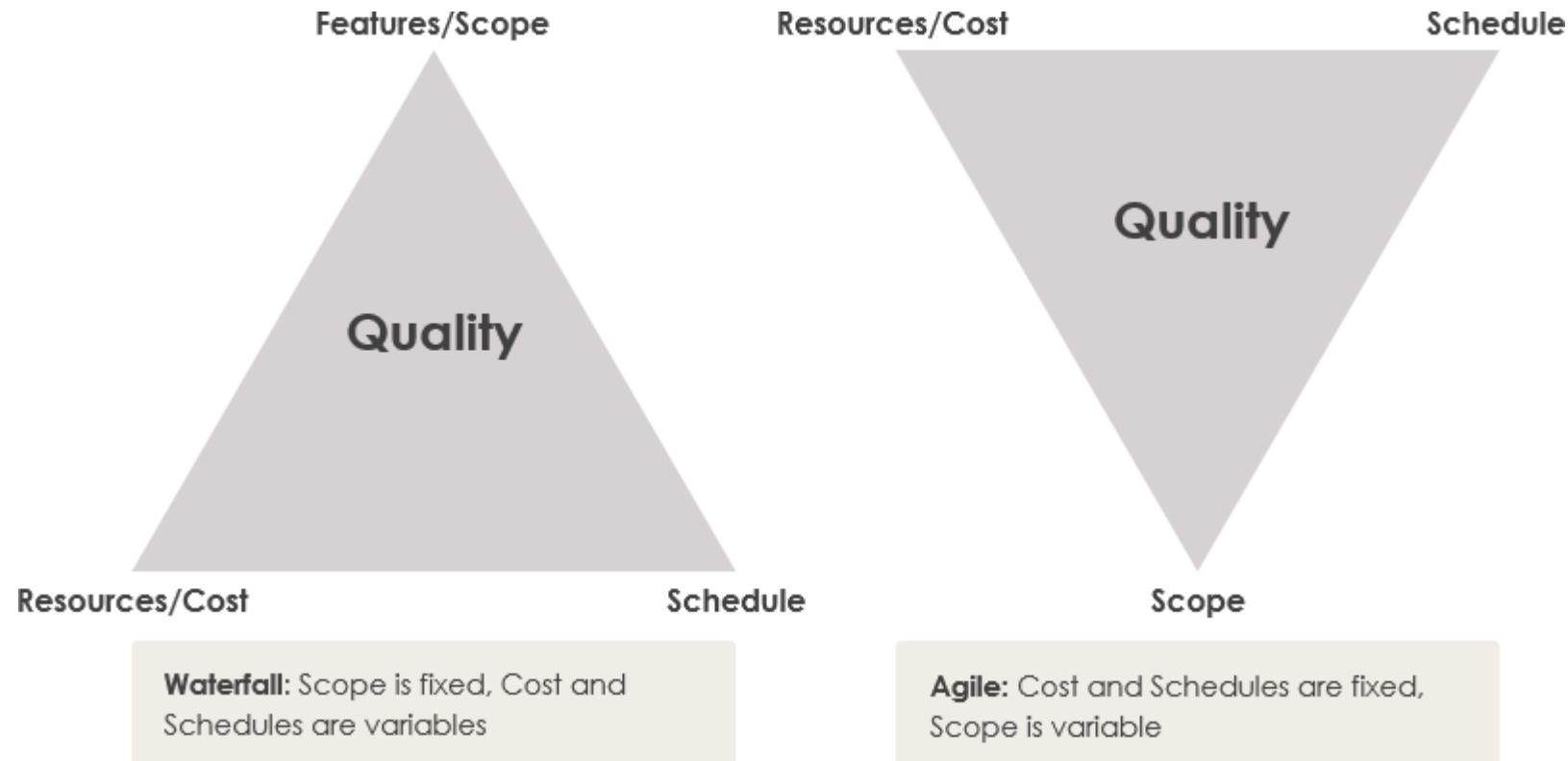
차세대 시스템 오픈에 따른 서비스 중단 - Google 검색



[Continuous Integration and Deployment Best Practices on AWS \(ARC307\) | AWS re:Invent 2013 \(slideshare.net\)](#)

Waterfall과 Agile의 비교

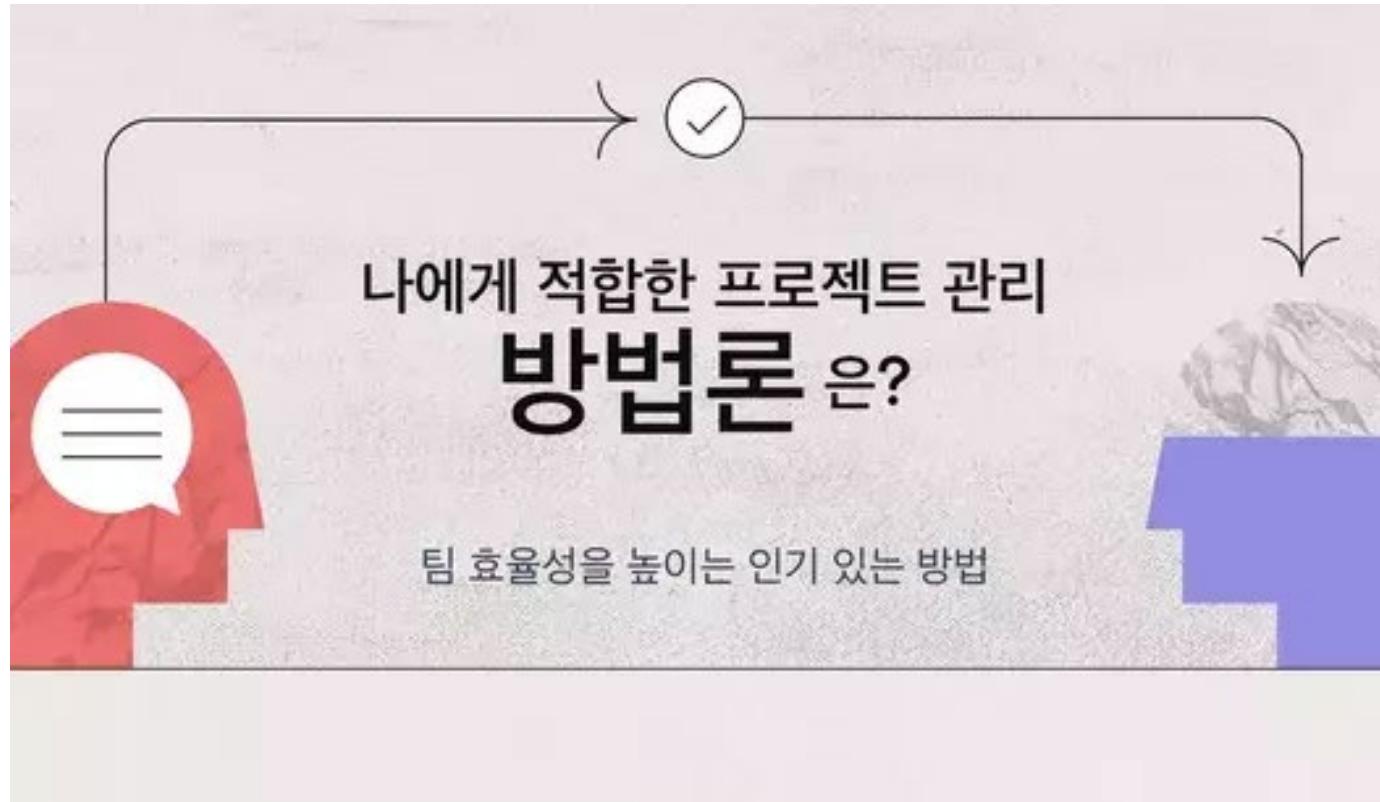
두 방법론은 어떤 요소를 고정시키고, 어떤 요소를 변화시키는가에 따라서 차이를 나타냅니다.



<https://www.visual-paradigm.com/scrum/scrum-vs-waterfall-vs-agile-vs-lean-vs-kanban/>

나에게 적합한 프로젝트 관리 방법론은?

Asana site 참고 (<https://asana.com/ko/resources/project-management-methodologies>)

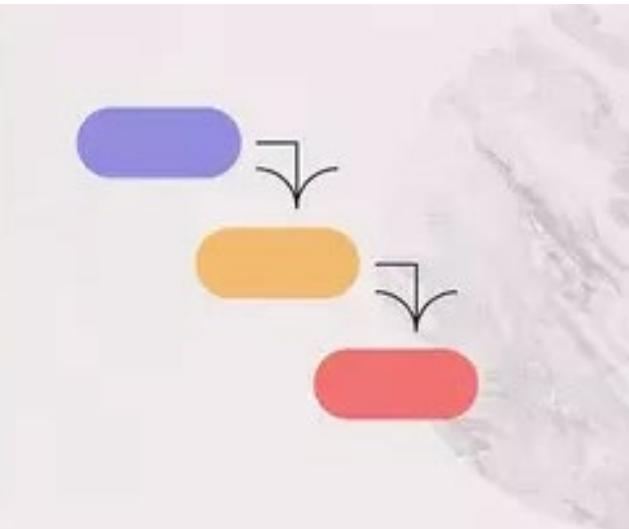


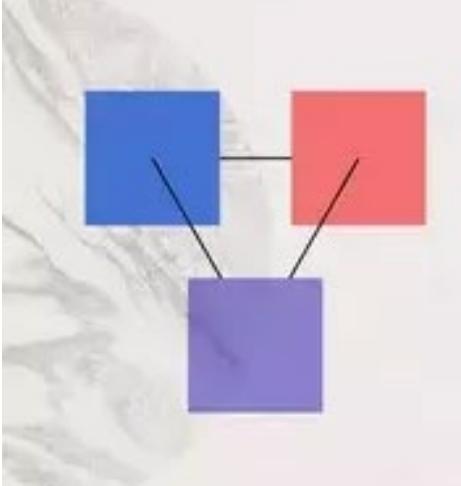
나에게 적합한 프로젝트 관리 방법론은?

워터폴

정의: 작업이 순차적으로 진행되는 선형 프로세스

적합한 팀: 단계와 작업 종속성을 명확히 하고 싶은 팀





PRINCE2

정의: 프로젝트 결함을 제거하기 위해 품질 관리에 사용되는 방법

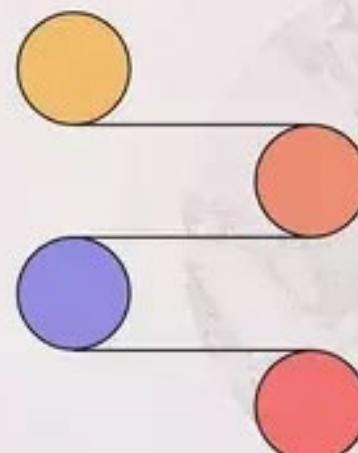
적합한 팀: 복잡한 프로젝트를 수행하는 대규모 팀

나에게 적합한 프로젝트 관리 방법론은?

린

정의: 낭비를 없애고 프로젝트 요구 사항에 맞는 간단한 프레임워크 작성을 목표로 하는 방법

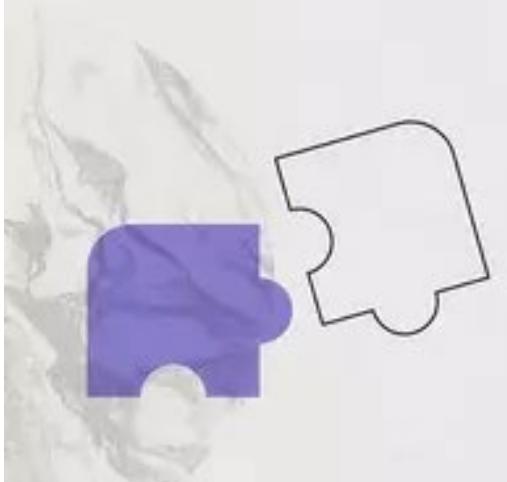
적합한 팀: 효율성 문제로 어려움을 겪고 있는 팀



스크럼

정의: 단기간의 "스프린트"를 반복하여 프로젝트 주기를 만드는 방법

적합한 팀: 단계적 접근 방식으로 분할하여 해결하려는 팀

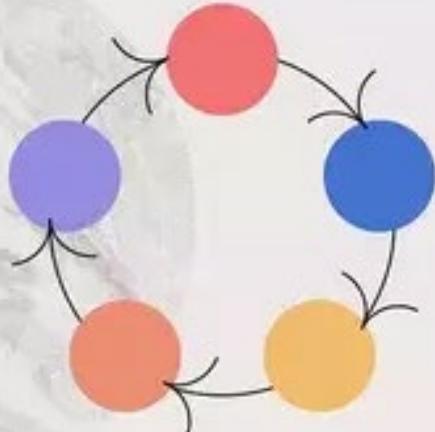
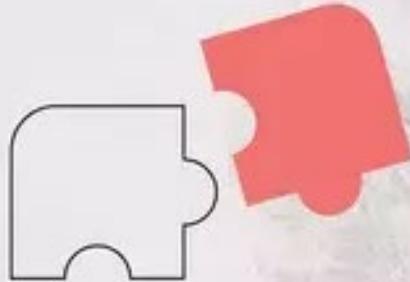


나에게 적합한 프로젝트 관리 방법론은?

칸반

정의: 보드와 같은 시각적 요소를 사용하여
프로젝트 백로그를 나타내는 방법

적합한 팀: 업무를 시각화하는 것을 선호하는 팀



식스 시그마

정의: 프로젝트 결함을 제거하기 위해 품질 관리에
사용되는 방법

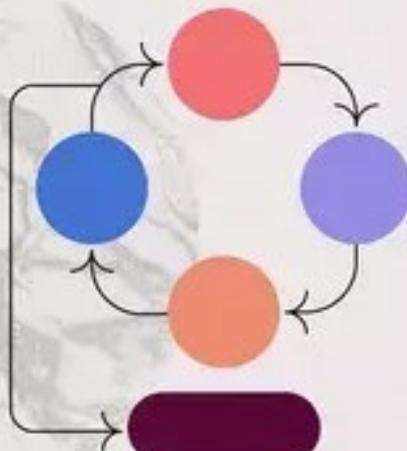
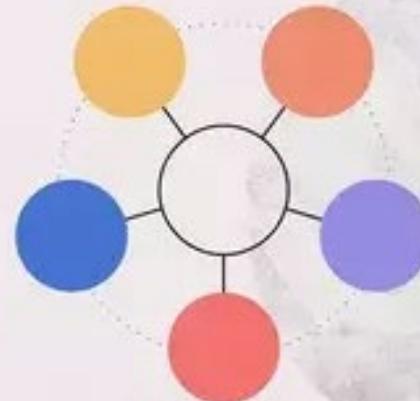
적합한 팀: 실수를 줄이려는 대규모 조직

나에게 적합한 프로젝트 관리 방법론은?

크리티컬 체인

정의: 작업 주제를 포함하고 업무를 목표와 연결하는 프로세스

적합한 팀: 시간 효율성 문제를 해결하는데 도움이 필요한 팀

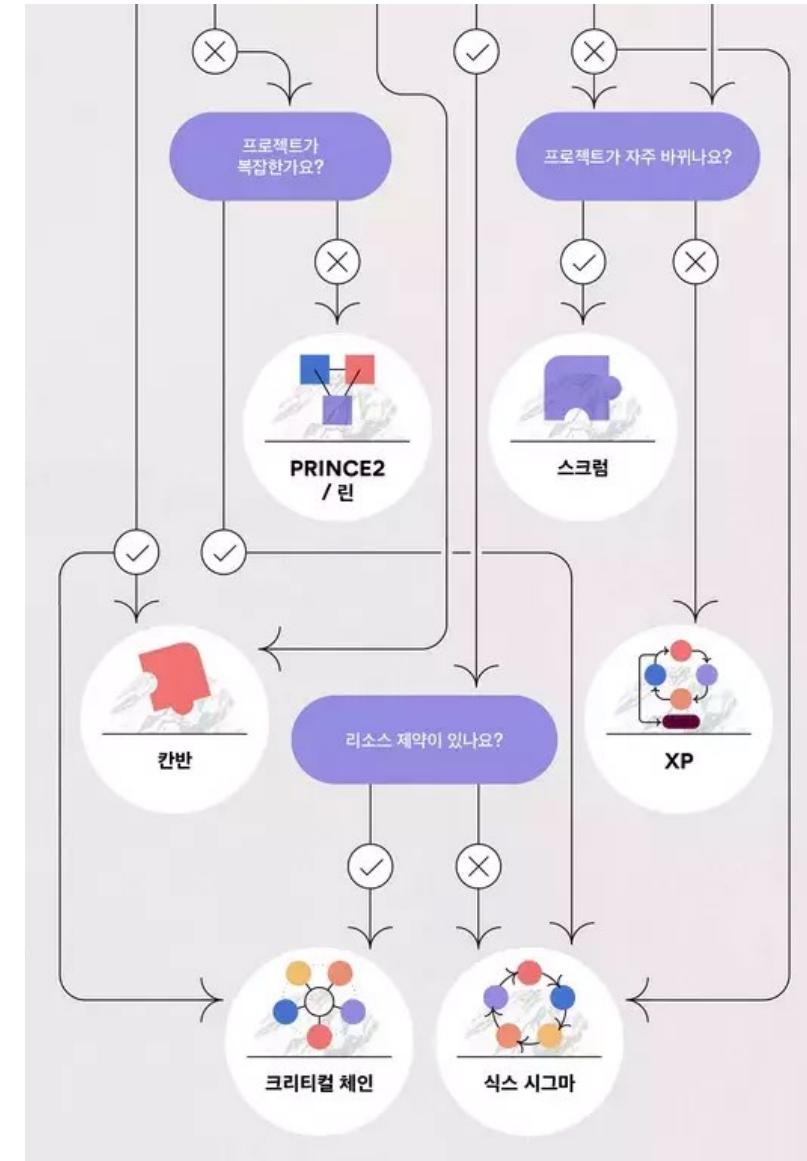


익스트림 프로그래밍

정의: 짧은 개발 주기로 많은 릴리스를 수행하는 방법

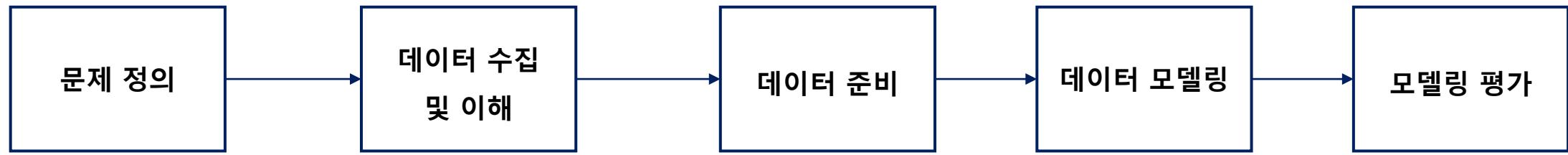
적합한 팀: 처리 시간이 측박한 중소 규모 팀

방법론 선택 흐름도



머신 러닝의 5단계

머신러닝을 이용한 시스템은 아래의 5단계를 거치면서 모델을 구현한다.



- | | | | | |
|--------------------------------|---------------------------------------|--|--|-------------------------|
| • 비즈니스 목적
정의 | • 데이터 수집
• 데이터 이해
• 데이터
레이블링 | • 데이터 전처리
• 데이터 분리
(학습 및
테스트) | • 모델링 알고리즘
선택
• 모델 훈련
• 모델 조정 | • 테스트
• 모델링 결과
평가 |
| • 환경/상황 분석 | | | | |
| • 모델링 목표
설정(유형,
알고리즘 검토) | | | | |

제안서 작성

제안서는 어떻게 작성해야 할까요?

제안서에서 강조해야 할 내용은 제안의 성격에 따라 다르게 됩니다만,
제안서를 구성하는 요소들을 어느 제안서나 크게 다르지 않습니다.

늘 6하원칙을 생각하는 습관
을 갖는 것이 좋습니다.

5W 1H

누가	Who
언제	When
어디서	Where
어떻게	How
무엇을	What
왜	Why

그 중에서도 가장 중요한 것
은 What, How, 그리고 Why
입니다.

How: 어떻게 할 것인가?
What: 무엇을 할 것인가?
Why: 왜 하는가?

제안서에서는 어떤 흐름으로
전개가 될까요?

어떤 과제를 하기 위해서는
'왜'가 제일 먼저 나와야 합
니다. 이 비즈니스를 하는 이
유는 무엇인가? 왜 해야 하
는가? 비즈니스의 정당성이
확보되면, 그 다음엔 무엇을,
어떤 것을 할 것인가?
그 이후에 어떻게 해야 잘 수
행할 수 있는가?의 흐름이
이어집니다. *

- 사업 수행시 기대 효과는 뒤에 다시 나오게 됩니다. '왜'에 해당되지만, 처음에 말하는 '왜'와는 달리 해당 과제 이후에 개선되고 기대할 수 있는 부분에 집중합니다.

RFP (Request for Proposal)

프로젝트의 owner는 프로젝트를 수행할 사업체를 선정함에 앞서서 제안서를 요청하게 되는데, 이때 작성하는 문서는 RFP입니다.

제안요청서

사업명	차세대 통합경영관리시스템 구축 용역
주관기관	한국원자력환경공단

2019. 10.

한국원자력환경공단
KOREA RADIOACTIVE WASTE AGENCY

담당	소속	직급	성명	전화번호	e-mail
예산정보팀	차장	임규혁	054)750-4122	gg1037@korad.or.kr	
예산정보팀	대리	이현민	054)750-4126	certllee@korad.or.kr	

I 사업개요

1. 개요

가. 사업명 : 차세대 통합경영관리시스템 구축 용역
나. 사업기간 : 청수일 ~ 2020.12.31. (약 14개월)
* 본 사업은 국가계약법 제69조에 의거하여 장기계약형 시행
* 상기 사업기간은 계약체결일, 조기구축, 혐의 등에 따라 변경될 수 있음
다. 사업금액 : 1,556,281,144 원(부가세포함)
라. 계약방법 : 제한경쟁입찰(협상에 의한 계약)

2. 추진배경

가. 본 사업 발주기관은 ERP를 이용하여 운영함에 따라 중복되는 기능 및 사용하지 않는 기능이 다수 있으며,
나. 관련 정보시스템, 설비의 노후화 및 각 ERP별 제조사 정책이 상이하여 개발·개선·데이터 연계의 어려움이 있는 상황임
다. 이에 차세대 그룹웨어 및 단일화된 ERP 도입을 통해 데이터 연계성, 무결성을 강화하여 업무효율성 향상을 목표로 함

3. 필요성 및 목적

가. 단일화 된 데이터 구조 구성을 통한 ERP, 각 정보시스템 간 데이터 연계
무결성 확보 필요
나. 차세대 통합경영관리시스템(ERP+그룹웨어) 구축을 통한 업무체계 확보 및 업무효율성 극대화를 통한 사용자 만족도 확보 필요
다. 4차 산업혁명 기반의 정보시스템, 기능 구축을 통한 미래지향적 경영체계 구축
라. 전자적 거래관리 및 연계 활용화를 통한 협업시스템 구축 필요
마. 공공기관에 특화된 ERP 및 그룹웨어를 도입하여 경부정책 등 대외환경변화를 정보시스템에 반영하는 등 유연한 정보시스템 및 설비 구축
바. 정보시스템 노후화 관점 개선 및 도입강비, 정보시스템 장기간 활용성을 고려하여 확장성이 뛰어나며 내구성 과학에 적합한 정보시스템 및 설비 구축
사. 범용성·확장성·가용성·신속성·무결성을 확보할 수 있도록 사용자 중심의 시스템 구축



제안요청서 예시

공공SW사업+제안요청서+작성을+위한+요구사항+상세화+실무가이드라인1.pdf

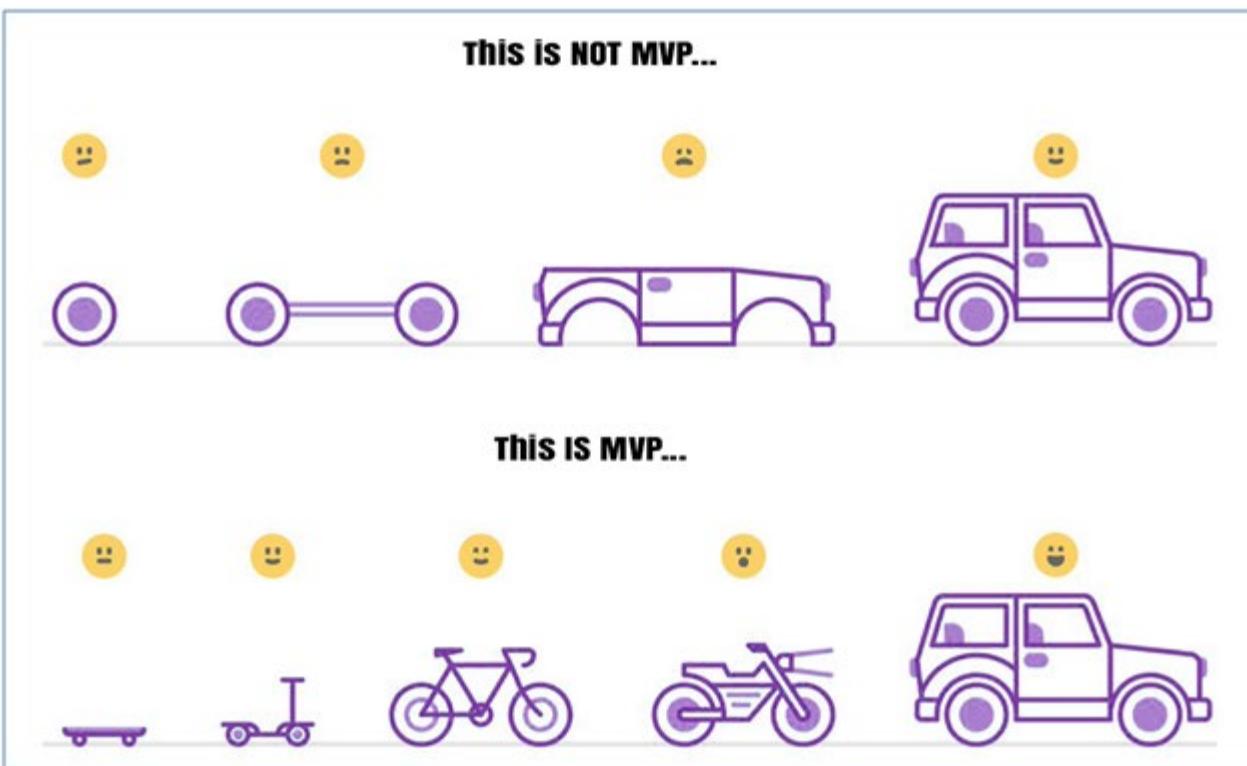
제안서 작성 사례

다음은 SI 프로젝트 제안서 목차 예시입니다. 모든 과제가 이렇게 복잡한 구성을 따를 필요는 없습니다. 참고만 하시기 바랍니다.

1. 제안 개요	제안 목적 제안 범위 전제 조건 제안사 특징 및 장점	프로젝트 관리 계획 품질 보증 계획 추진 일정 계획 보고 및 검토 계획 일정 관리 방안 개발 산출물 관리 방안 산출물 형상 관리 방안 이슈 및 위험 관리 보안 관리 수행 조직 및 업무 분장 투입 인력 및 이력 교육 훈련 및 계획 유지보수 및 기술이전 계획 운영효율화 전략 기타 지원 사항
2. 제안 업체 일반	일반 현황 조직 및 인원 주요 사업 내용 주요 사업 실적 협력사 주요 현황	4. 프로젝트 관리 부문
3. 프로젝트 수행전략 부문	차세대 구축 방안 시스템 구축 방안 시스템 아키텍쳐 제안	5. 지원부문
6. 기타	인력 투입 계획 사업 수행 일정	

MVP (Minimum Viable Products)

The term was coined and defined in 2001 by Frank Robinson[4] and then popularized by Steve Blank and Eric Ries (Lean Startup)



How **not to build** a Minimum Viable Product



How **to build** a restaurant like business with Minimal Viable Product

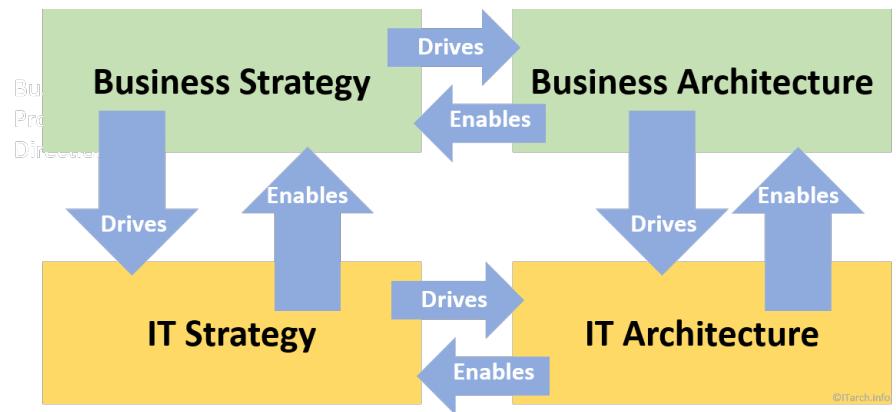
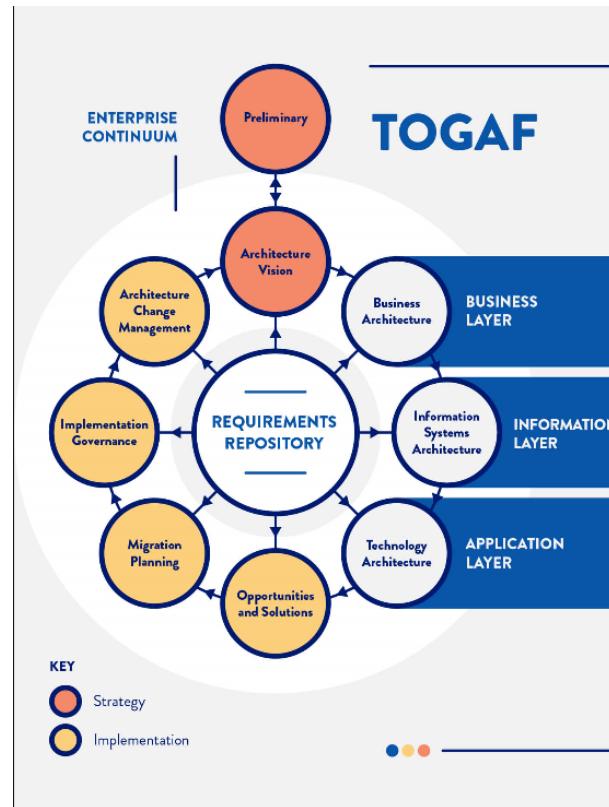
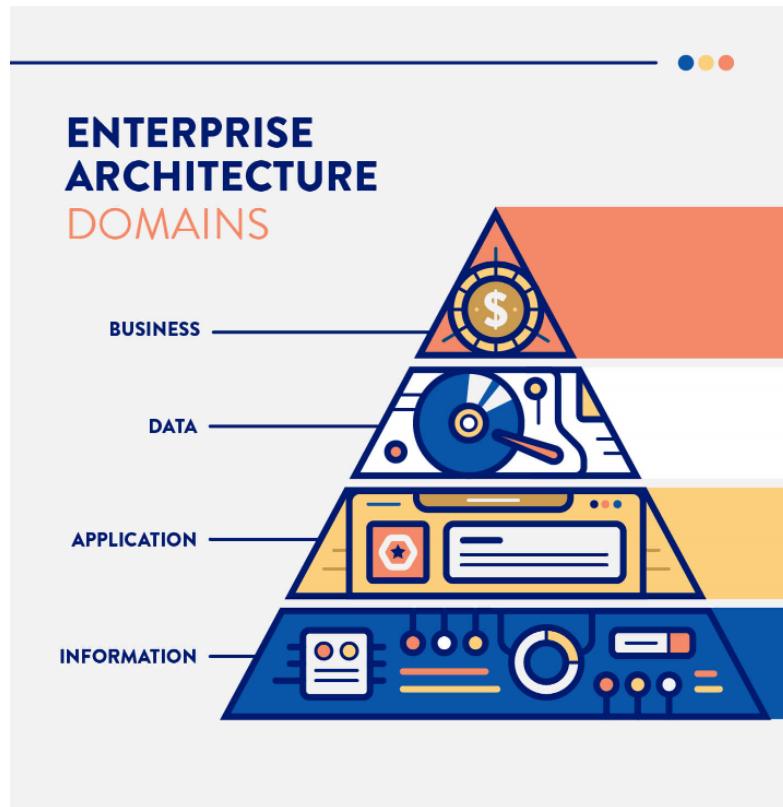


How **to build** a software product with Minimal Viable Product



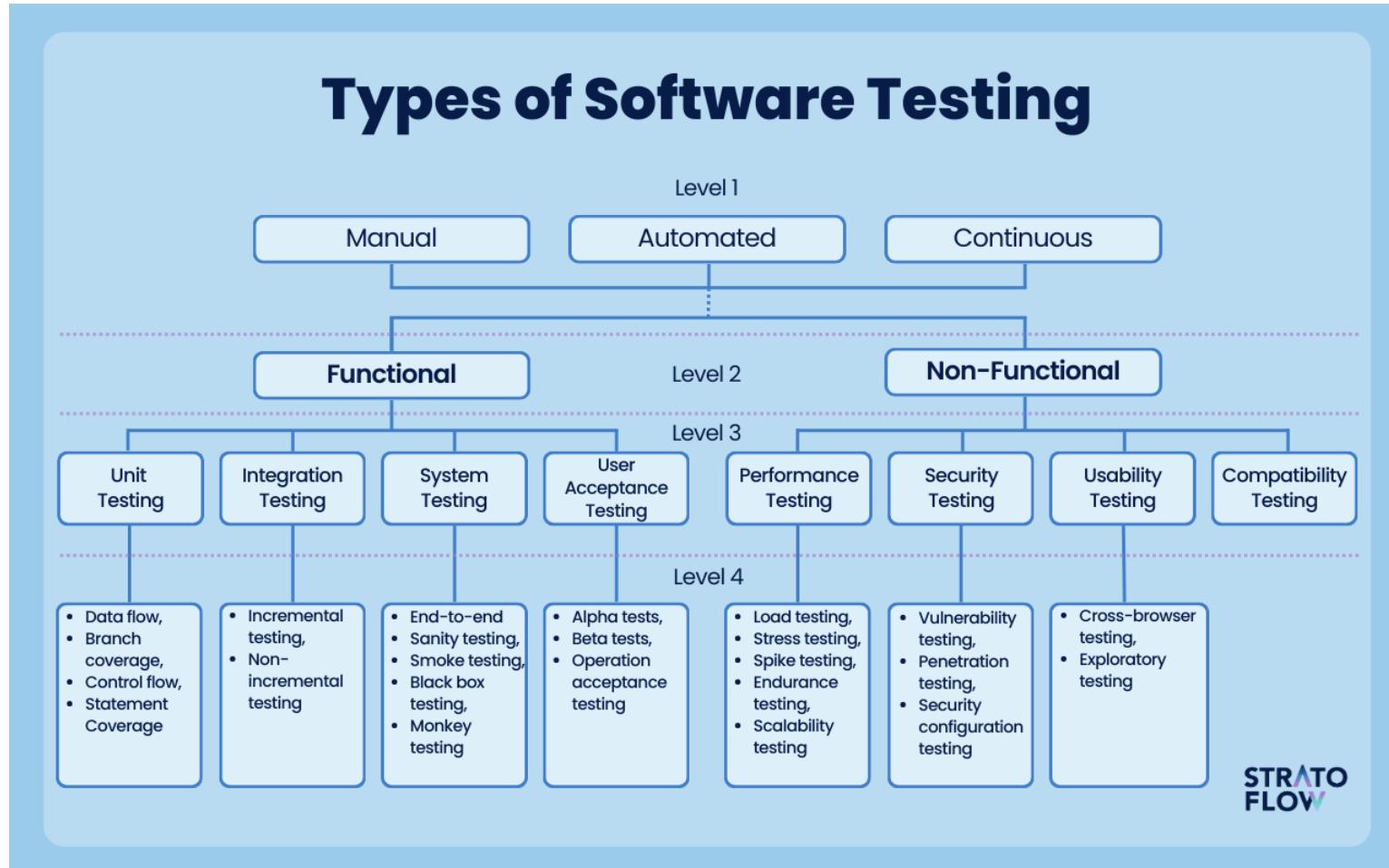
Enterprise Architecture

엔터프라이즈 아키텍처는 기업의 비즈니스와 IT 운영을 일관성있게 전략적으로 행동하기 위해 정의하는 것입니다. 제안서에서 Why와 What이 연결되듯이, 비즈니스와 IT가 연결된다는 생각을 항상 갖도록 노력하라는 의미로 참고해볼만 합니다. (단, EA 자체는 상당히 광범위해서 바로 적용하기에는 무리가 많습니다.)



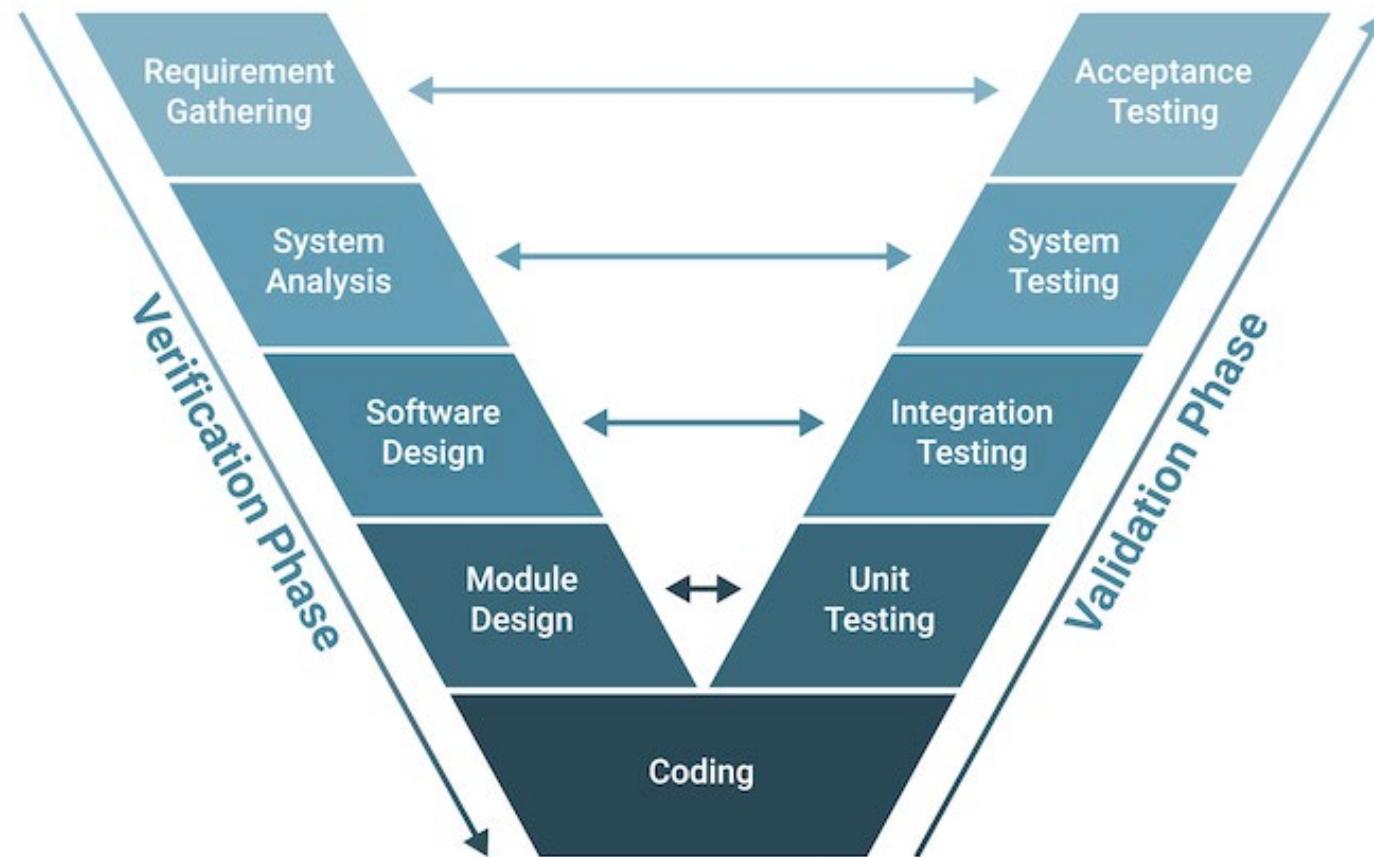
테스트의 종류

소프트웨어 테스트는 실행 방법에 따라, 성격에 따라, 단계에 따라 아래와 같이 나눠볼 수 있습니다.



V-model

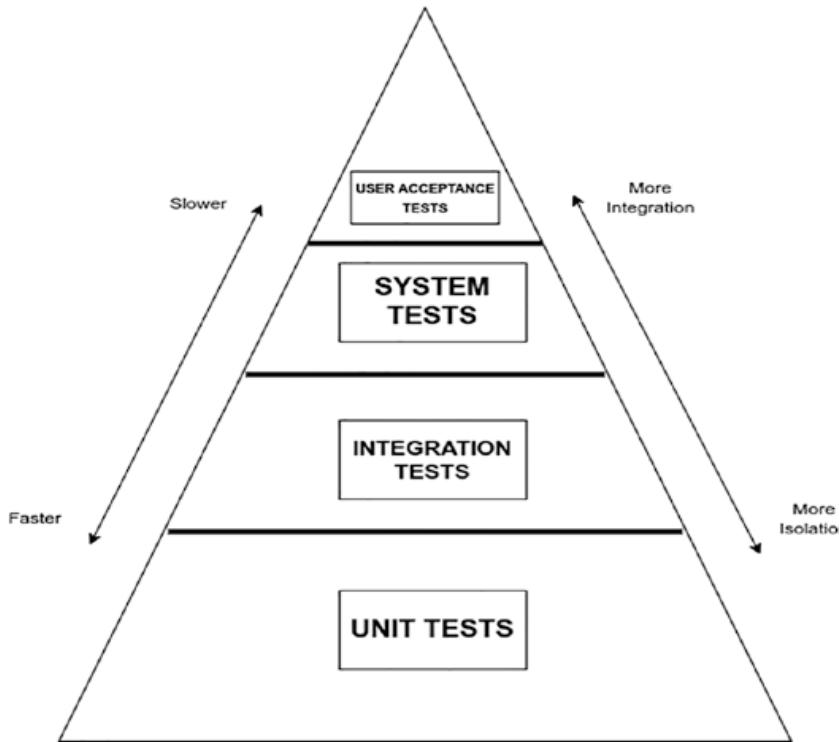
V-model은 아래 모양에서 볼 수 있듯이 소프트웨어 개발 라이프사이클을 V로 표시한 모델입니다. 왼쪽은 코딩 이전 분석, 설계 단계, 코딩 이후는 단계별 테스트를 보여줍니다.



<https://builtin.com/software-engineering-perspectives/v-model>

Test Pyramid

Mike Cohn*이 Test Pyramid라는 아이디어를 제시한 이후 다양한 테스트 피라미드 모델이 제시되었고, 아래는 그 중 하나입니다. 보는 시각에 따라 다양한 피라미드 모델이 존재하지만, 동일한 점 하나는 아래로 갈수록 테스트 시간이 짧고, 독립적으로 진행되고, 위로 갈수록 테스트 시간이 길고 통합되어 진행되어야 한다는 점입니다.

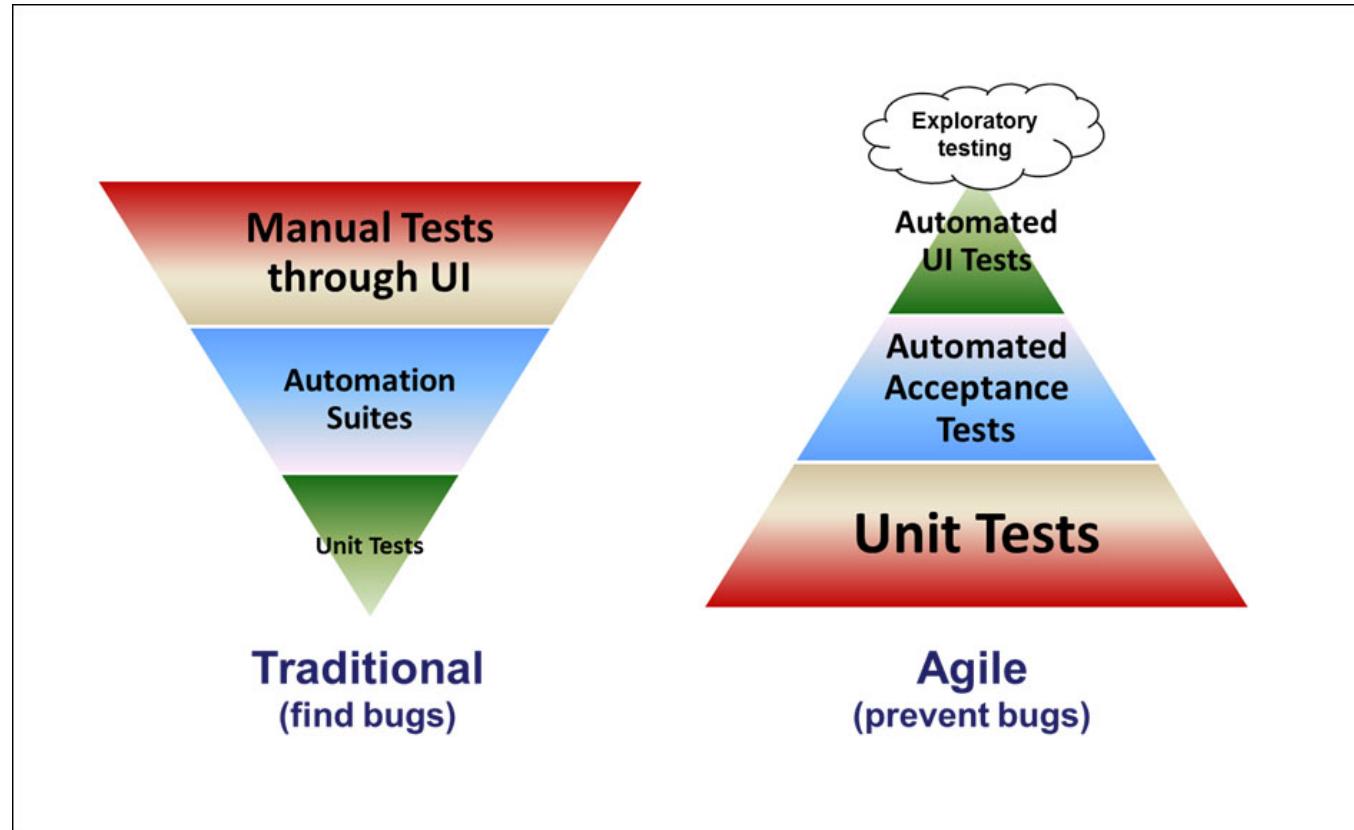


Introduction to Software Testing, Panagiotis Leloudas, Apress

* Mike Cohn은 Agile 소프트웨어 개발에 대한 교육 및 컨설팅을 제공하는 회사를 운영하고 있습니다.

Traditional vs Agile

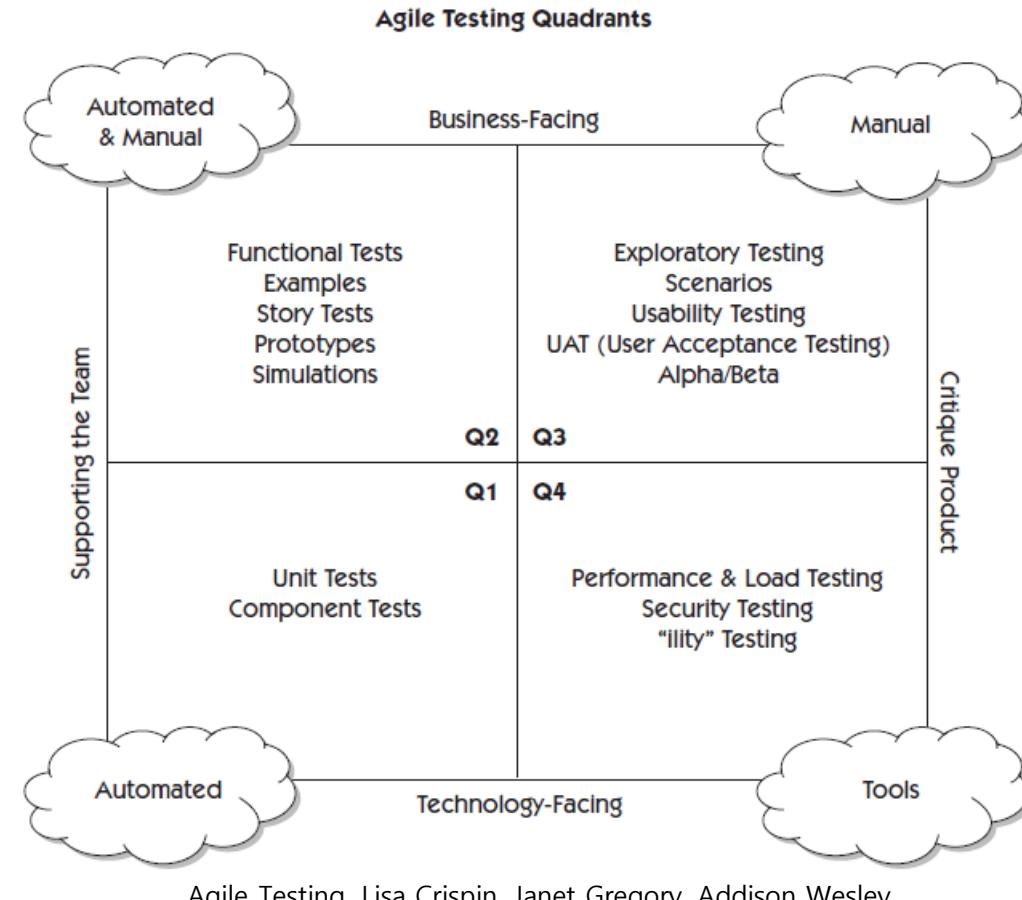
Agile 방법론의 특징 중 하나는 테스트를 강조한다는 것이고, 그 이유는 아래 그림을 통해 설명될 수 있습니다.



<https://www.agilecoachjournal.com/2014-01-28/the-agile-testing-pyramid>

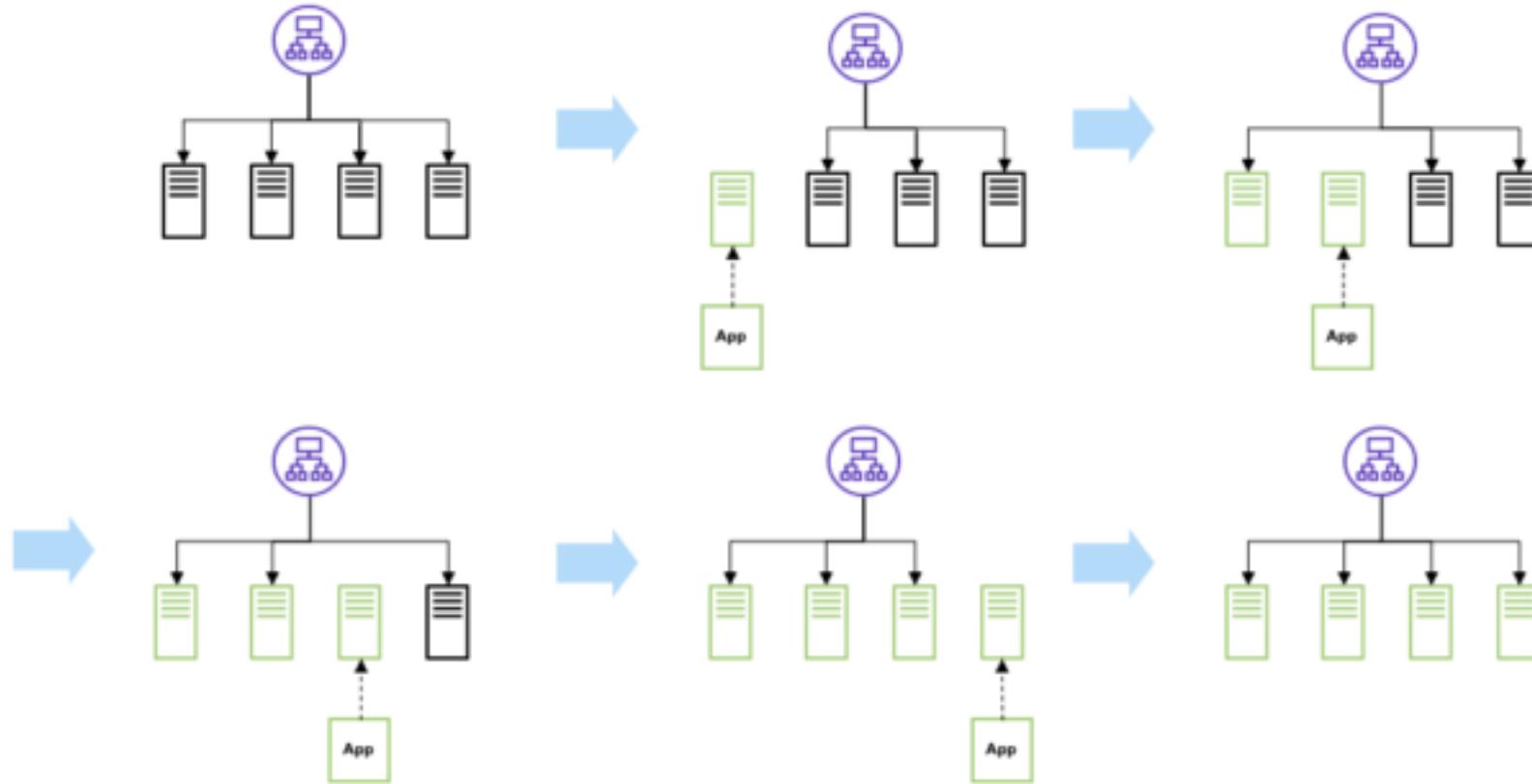
Agile Testing Quadrants

Agile 접근방법에서는 테스트의 목적, 성격 등에 따라 아래의 quadrants 그래프 영역에 테스트를 분류하고 있습니다.



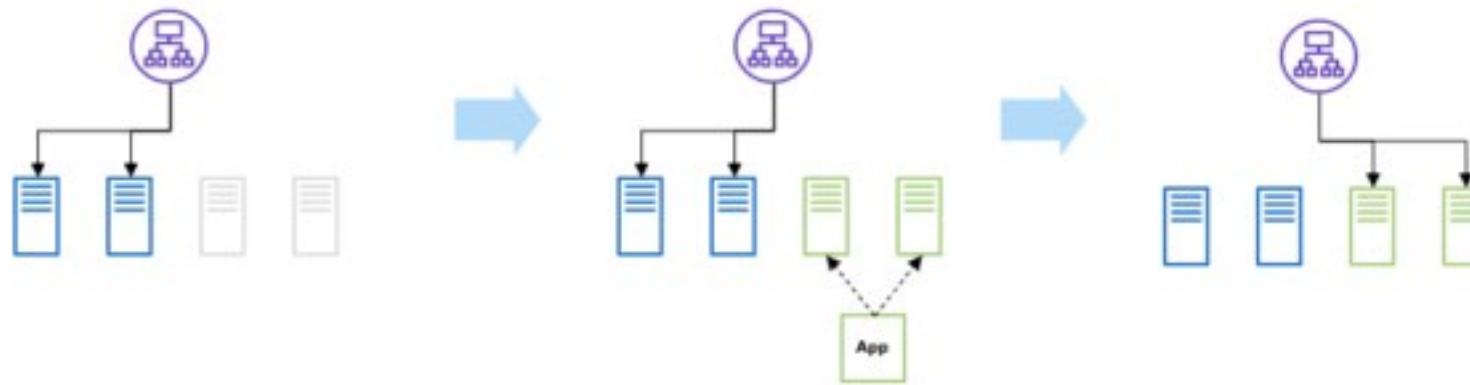
배포 방안

1. 롤링 배포(Rolling Update Deployment)



배포 방안

2. 블루/그린 배포(Blue/Green Deployment)



배포 방안

3. 카나리 배포(Canary Deployment)

