



7

적응형 라이프 사이클(II)

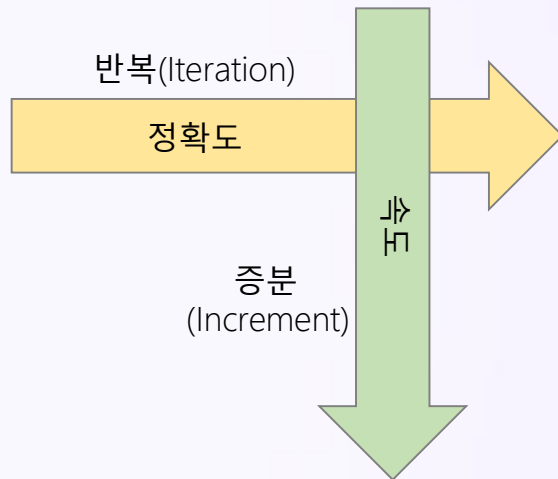
오늘의 학습내용

- 애자일 라이프 사이클(Agile Life Cycle)의 특징
- Iteration-Based Agile vs Flow-Based Agile

▲ 애자일 라이프 사이클(Agile Life Cycle)

✕ ✕ 애자일 라이프 사이클(Agile Life Cycle)이란?

- = 적응형 라이프 사이클
(Adaptive Life Cycle)
- = 변경주도형 라이프 사이클
(Change-driven Life Cycle)
- = 반복적 및 점증적 라이프 사이클
(Iterative and Incremental Life Cycle)
- = 반복(Iteration) + 증분(Increment)
- = 스크럼(Scrum)
- = 점진적 상세화(Progressive Elaboration)



▲ 애자일 라이프 사이클(Agile Life Cycle)

애자일 라이프 사이클(Agile Life Cycle)은 반복(Iteration)과 증분(Increment)에 따라 프로젝트를 점진적 상세화(Progressive Elaboration)하는 것을 의미함

반복(Iteration) = 표준 프로세스 = 단계(Step, Stage, Phase)

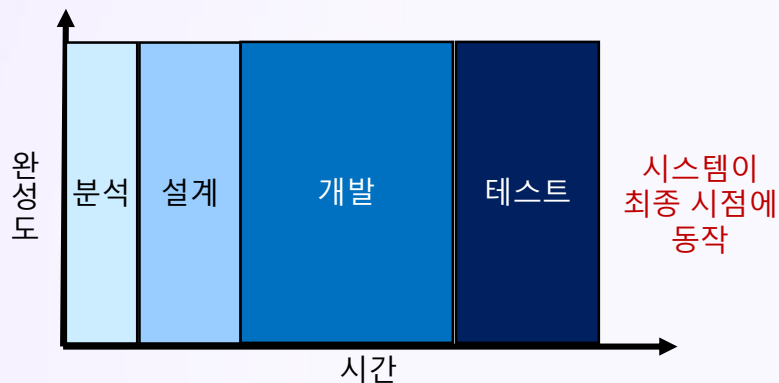
- 제품 개발을 위한 프로젝트 단계를 개발함(가로로 프로세스 적용)
- 반복 프로세스의 마지막 작업인 인도(Delivery) 전에 산출물이 완성되어야 함

증분(Increment) = 점증 = Build = Version

- 제품의 기능을 추가하거나 보완함(세로로 프로세스 적용)
- 증분에 의해 프로젝트를 지속해 나감
 - 프로젝트 초기에 상위 수준의 범위가 명확하지만, 상세한 범위는 증분을 반복하면서 구체화됨
 - 현재의 증분에서 다루지 않는 산출물은 상위 수준에서 다루거나 미래의 증분N에 잠정적으로 배분함

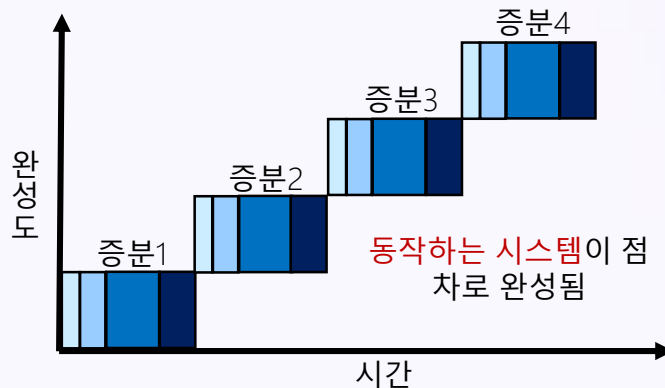
▲ 애자일 라이프 사이클(Agile Life Cycle)

■ 예측형(Predictive) vs 애자일(Agile)



예측형 라이프 사이클(Predictive Life Cycle) = 폭포수 모델(Waterfall Model)

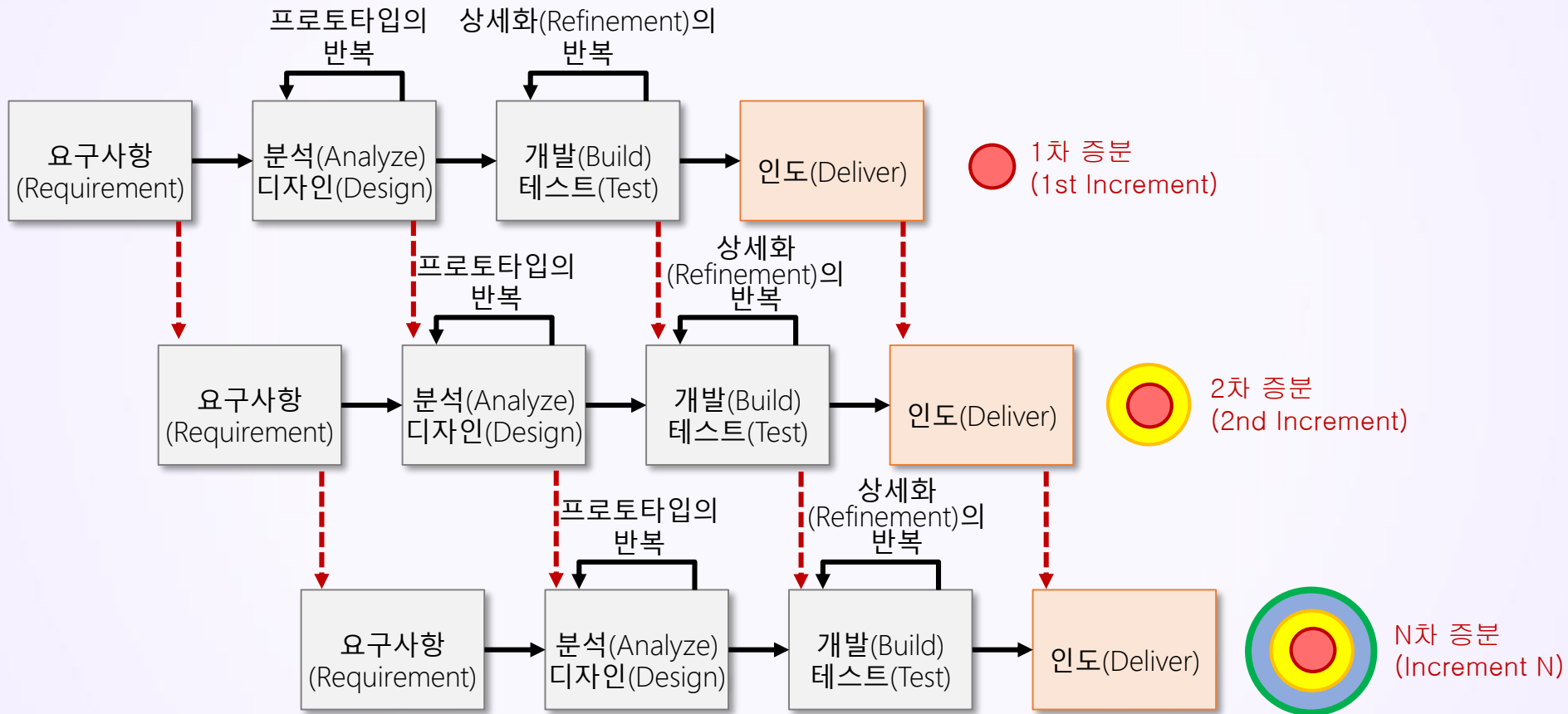
- 분석(Analysis): 무엇을 만들 것인지 이해
- 설계(Design): 어떻게 만들지 검토
- 개발(Development): 실제 소프트웨어 작성
- 테스트(Test): 만든 것을 확인하고 검증



애자일 라이프 사이클(Agile Life Cycle) = 적응형 라이프 사이클(Adaptive Life Cycle)

- 증분1(Increment 1): 최소한의 핵심 기능을 개발
- 증분2(Increment 2): 증분1이 발전 → 사용자의 관심과 우선순위가 높은 기능을 개발
- 증분3(Increment 3): 증분2가 발전 → 그 다음 우선순위가 높은 기능을 집중 개발
- 증분4(Increment 4): 증분 3이 발전 → 남은 기능을 개발

▲ 애자일 라이프 사이클(Agile Life Cycle)



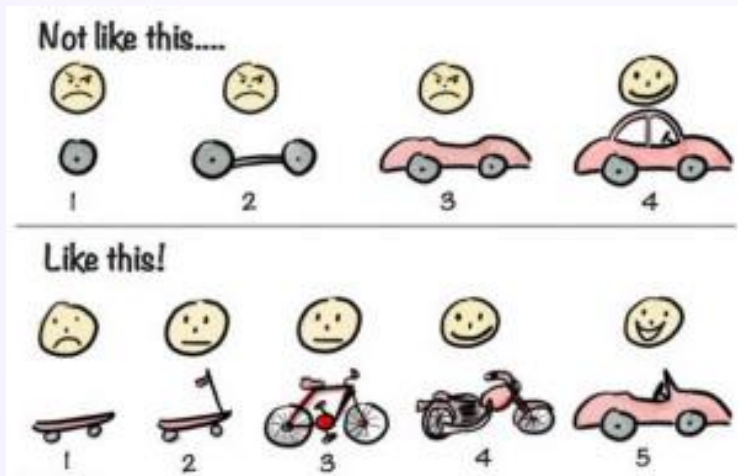
▲ 애자일 라이프 사이클(Agile Life Cycle)

애자일 라이프 사이클(Agile Life Cycle)은 반복(Iteration)과 증분(Increment)에 프로젝트를 실행함



증분(Increment)은

작은 사이클을 반복하여 최소 기능 제품
(MVP: Minimum Viable Product)을 진화시켜 나가는 과정



- 애자일 라이프 사이클(Agile Life Cycle)을 프로젝트에 적용하면 MVP를 일찍 받아 볼 수 있으며, 피드백을 통해서 더 훌륭한 모습으로 다음 제품을 만들 수 있음
- 애자일 라이프 사이클은 증분(Increment) 기간이 짧고 계속 진화하여 나감

Source: Henrik Kniberg

<https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp>

▲ 애자일 라이프 사이클(Agile Life Cycle)

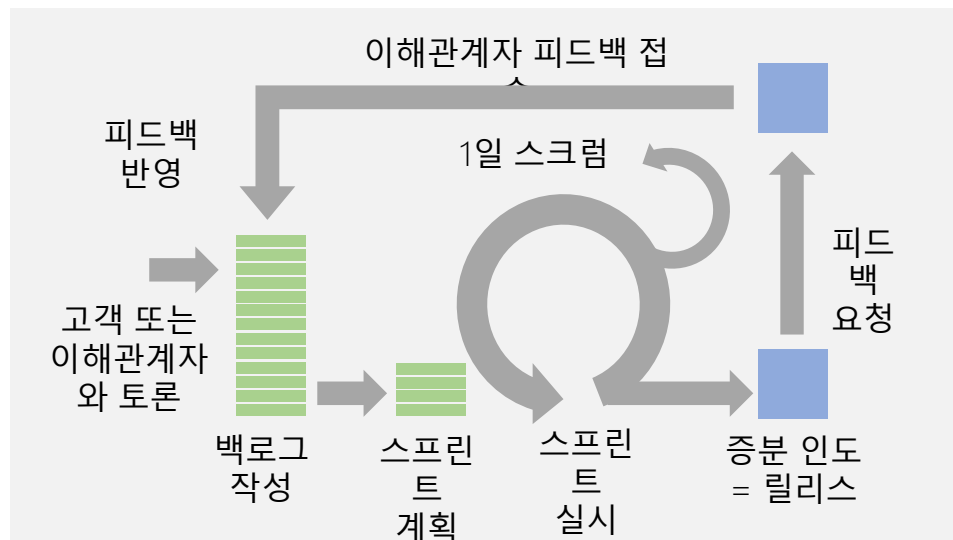
애자일 라이프 사이클(Agile Life Cycle)의 목표

반복(Iteration)의 가치 실현

- 프로젝트 범위의 변경을 관리하여 불확실성을 줄임
- 다양한 프로세스를 활용함
 - 예 개발 프로세스, 변경 통제 프로세스, 리스크 관리 프로세스 등

증분(Increment)의 가치 실현

- 여러 번의 인도와 피드백을 통한 고객 가치(Customer value via frequent deliveries and feedback)를 제공함
- MVP와 사업적 가치(Business Value)에 도움이 되는 중요한 것부터 먼저 하며, 빠르게 인도(Deliver Fast)함
- 불필요한 개발로 인한 낭비를 줄일 수 있으며 원가 절감에 할 수 있음



▲ 애자일 라이프 사이클(Agile Life Cycle)

✕✕ 애자일 라이프 사이클(Agile Life Cycle)을 위한 방법론 ✕✕

스크럼(Scrum)

반복(Iteration)과 증분(Increment)에 따라
개발 일정을 진행함

칸반(Kanban)

스프린트 내에서의 계획(Plan)과 진행 상황(Progress)을 시각화하고
종합적인 관점에서 프로젝트 현황을 분석함

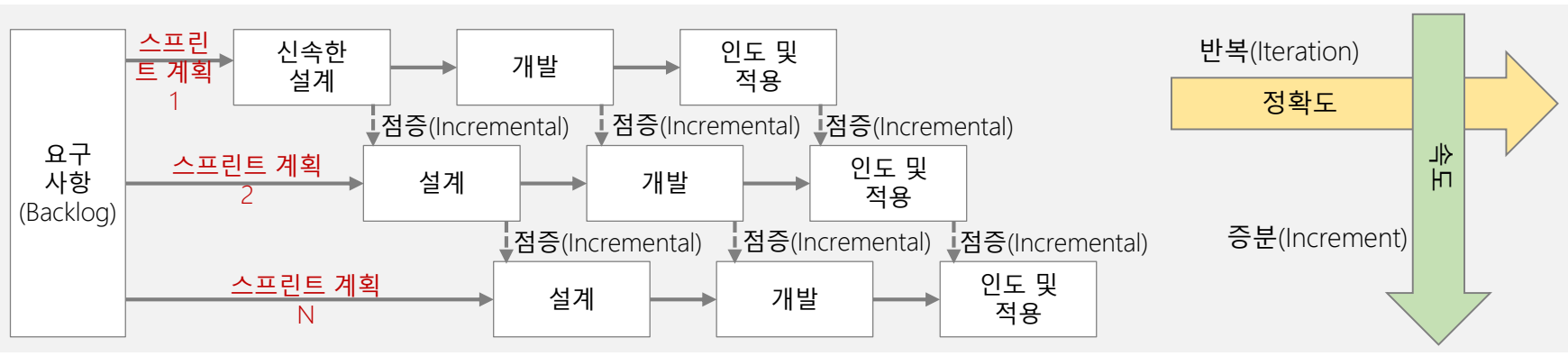
익스트림 프로그래밍(eXtreme Programming)

개발자가 주도하여 범위를 결정하고 개발을 진행하며 창의적으로
문제를 해결하는 다양한 기법을 실천함

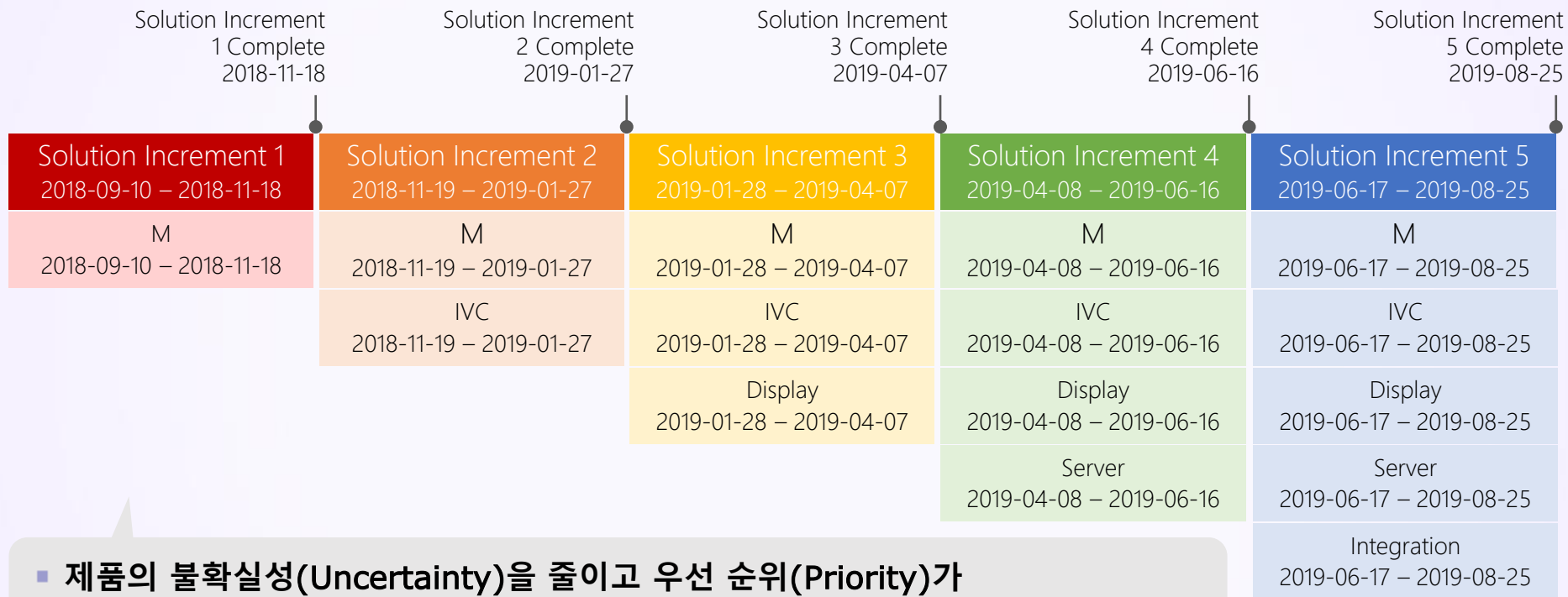
▲ 애자일 라이프 사이클(Agile Life Cycle)

애자일 라이프 사이클(Agile Life Cycle)의 특징

- 빠르고(Agile), 반복적이며(Iterative) 또는 점증적(Incremental)임
- 제품의 부분적 인도에 대한 대가가 있을 때 필요한 개발 모델임
- 애자일 라이프 사이클(Agile Life Cycle)에서는 반복을 시작하기 전에 자세한 범위를 정의하고 승인해야 함
(The detailed scope is defined and approved before the start of an iteration)
- 일단 반복 프로세스를 착수하면 범위에 대한 변경을 엄격하게 통제해야 함
(Changes to the scope of an iteration are carefully managed once work begins)
- 증분의 수가 많고 병행 개발이 빈번하게 이루어지면, 관리가 어려워지고 증분 활동 간의 조율에 많은 노력이 필요함



▲ 애자일 라이프 사이클(Agile Life Cycle)



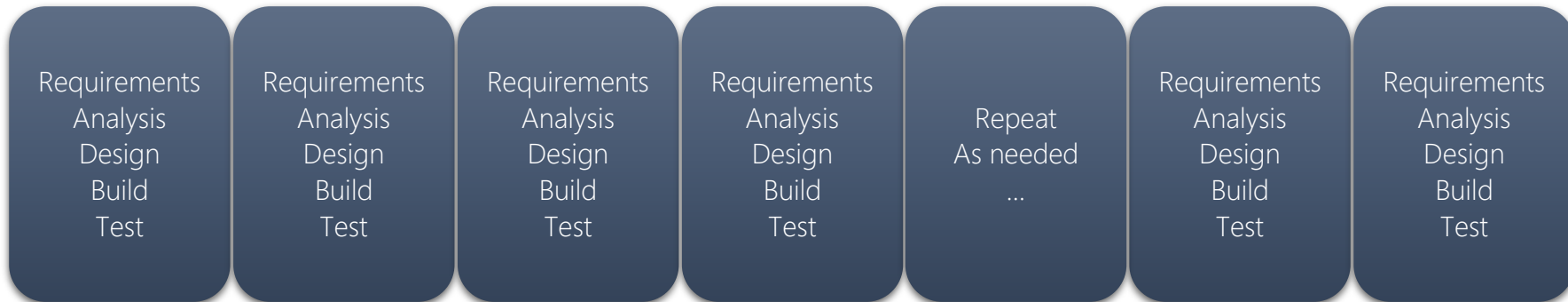
- 제품의 불확실성(Uncertainty)을 줄이고 우선 순위(Priority)가 높은 증분(Increment)을 먼저 개발하는 방식의 단계를 제시해야 함
- 스프린트(Sprint)를 설정하고 반복(Iteration)에 따라 각 증분(Increment)을 개발하고 인도함

▲ 애자일 라이프 사이클(Agile Life Cycle)

Iteration-Based Agile vs Flow-Based Agile

반복 기반(Iteration-Based) 애자일 라이프 사이클

- Iteration-Based Agile 라이프 사이클은 표준 프로세스를 주기적인 Time Box 내에서 진행하는 것을 의미함
- Increment(또는 Sprint) 목표를 제시하고, 각 Increment 내에서 표준 프로세스 계획을 수립함
- 표준 프로세스예 요구사항, 분석, 설계, 구축, 테스트
- Time Box, 즉 보고 주기를 2주로 하기로 했다면, Time Box 마지막 시점에 중간 산출물을 인도하는 것을 원칙으로 함



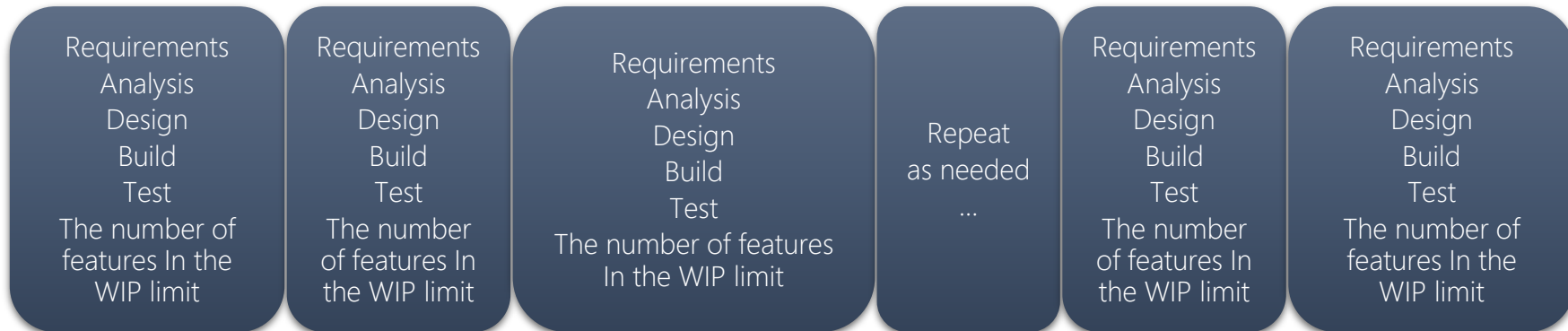
NOTICE : Each timebox is the same size. Each timebox results in working tested features.

▲ 애자일 라이프 사이클(Agile Life Cycle)

Iteration-Based Agile vs Flow-Based Agile

플로우 기반(Flow-Based) 애자일 라이프 사이클

- Flow-Based Agile 라이프 사이클은 요구사항에 따라서 Time Box를 각각 다르게 설정함
- 요구사항의 구현을 위하여 높은 창의력이 필요하거나 리스크 대응이 필요하다면 Time Box를 길게 잡을 수도 있음
- Flow-Based Agile 라이프 사이클에서는 Time Box가 과도하게 길어지는 것을 막기 위해서 WIP(Work In Progress) Limit을 설정함



NOTICE : In flow, the time it takes to complete a feature is not the same for each feature.

▲ 애자일 라이프 사이클(Agile Life Cycle)



Flow-Based Agile 라이프 사이클에서는

Time Box가 과도하게 길어지는 것을 막기 위해서 WIP(Work In Progress) Limit을 설정함

WIP(Work In Progress) Limit의 의미

- 진행 중인 작업의 개수를 제한하는 것. 할 수 있는 작업의 최대 개수
- 우선순위가 높은 작업들을 먼저 Task Board(Kanban)에 등록함
- WIP Limit 개수를 넘어서면, 해당 워크플로우에 작업을 추가할 수 없음

WIP Limit의 목적 ⇒ Multi-tasking 지양

WIP Limit의 예

- Ready의 WIP Limit : 8개
- Development and unit test의 WIP Limit : 3개
- User Acceptance Test의 WIP Limit : 2개



KEY POINT

■ 애자일 라이프 사이클(Agile Life Cycle)

- 애자일 라이프 사이클(Agile Life Cycle)은 반복(Iteration)과 증분(Increment)에 프로젝트를 실행합니다.
- 증분(Increment)은 작은 사이클을 반복하여 "최소 기능 제품"(MVP: Minimum Viable Product)을 진화시켜 나가는 과정입니다.

■ 애자일 라이프 사이클(Agile Life Cycle)의 종류

- 애자일 라이프 사이클(Agile Life Cycle)에는 Iteration-Based와 Flow-Based의 두 가지가 있습니다.
- 반복 기반(Iteration-Based) 애자일 라이프 사이클은 표준 프로세스를 주기적인 Time Box 내에서 진행하는 것을 의미합니다.
- 플로우 기반(Flow-Based) 애자일 라이프 사이클은 요구사항에 따라서 Time Box를 각각 다르게 설정하며, Time Box가 과도하게 길어지는 것을 막기 위해서 WIP(Work In Progress) Limit을 설정합니다.