

2019311801 이균서

A1

- iterative model: 처음의 requirement 가 구체적이지 않아, outline 에서 시작을 한다. Specification - Design - Validation 과정이 반복이 된다. 그래서 상대적으로 Specification 비용이 준다. 대신 Waterfall 의 Integration 부분이 Iterative Development 로 들어가면서, cost 가 증가한다. 그래서 waterfall 의 cost distribution 과 다르다.
- component-based model: scratch 부터 software 를 개발하는 것이 아닌, reuse 를 통해 software 를 개발한다. 그러므로 integration 과 testing 의 비중이 높아진다.

A2

사실 어떤 두 attributes 의 combination 도 가능하다. 한 가지 예시를 들겠다. Efficiency 를 높이기 위해서는 Maintainability 가 일정부분 희생이 가능하다. 그 반대도 성립한다.

A3

Validation 은 고객의 니즈가 meet 이 되는지를 보는 것이고, Verification 은 Specification 을 통해 명세화된 문서와 일치하는지를 보는 것이다.

A4

- Specification: 소프트웨어의 기능 및 제약 조건을 정의합니다.
- Development: 소프트웨어 설계 및 프로그래밍을 포함합니다.
- Validation: 소프트웨어가 사용자의 요구 사항을 충족하는지 확인합니다.
- Evolution: 소프트웨어를 변경하여 새로운 고객 및 시장의 요구 사항을 충족하거나 변경된 환경에서 동작하도록 합니다.

A5

- Stand-alone applications: OS 내장 계산기 application, OS 내장 text editor
- Interactive transaction-based applications: 온라인 쇼핑몰, ATM 시스템
- Embedded control systems: 세탁기 시스템, 냉장고 시스템
- Batch Processing systems: 스마트폰 이용 요금 계산 시스템, 전기/수도 이용 요금 계산 시스템
- Entertainment systems: MMORPG 게임, 음악 플랫폼
- Systems for modelling and simulation: 미사일 발사 시뮬레이션, 전투기 이착륙 시뮬레이션 시스템
- Data collection systems: 날씨 정보 수집 시스템, 스마트폰 OS 사용자 정보 수집 시스템
- Systems of systems: 지구상의 대부분 시스템들

A6

고객의 요구 사항과 개발팀이 처한 상황이 다르기 때문이다.

A7

- 시스템은 이해하고 관리하는 개발 프로세스를 사용하여 개발되어야 합니다.
- 모든 유형의 시스템에 대한 신뢰성과 성능은 중요합니다.
- 소프트웨어 사양 및 요구 사항 (소프트웨어가 수행해야 할 작업) 의 이해와 관리는 중요합니다.
- 적절한 경우 이미 개발된 소프트웨어를 재사용하는 것이 좋습니다.

A8

시스템을 모아서, 재조립한다. (가져다 쓸 수 있는 건, 다 가져다 쓴다.)

A9

최근 AI 학습 model 에 저작권이 걸린 예술작품들을 무단으로 학습시켜 논란이 되고, 이것이 저작권과 관련된 법적인 문제까지 갔다.

A10

계속 새로운 것들을 학습하려는 열정이 중요한 것 같다.

Q1 from Student

회사마다 다른 개발 문화는 어디서 기인을 하는 것인지?

A

waterfall, agile 과 같은 다른 개발 방법론에서 기인하는 것 같다.

Q2 from Student

수 많은 개발 방법론 사이에 우열이 있는지?

A

각 팀이 처한 상황에 따라 다른 것이다. 그것에 맞게 선택하는 것이 중요하다. 상황에 맞는 선택은 우열이 존재할지도 모른다.