

Software Architecture 4+1 View에 대해서 설명해줘

소프트웨어 아키텍처 4+1 뷰 모델은 복잡한 소프트웨어 시스템의 아키텍처를 다양한 이해관계자의 관점에서 설명하기 위한 프레임워크입니다. 이 모델은 4개의 주요 뷰와 검증을 위한 시나리오 뷰(+1)로 구성되며, 각 뷰는 특정 목적과 대상에 초점을 맞춥니다.

주요 구성 요소

1. 논리 뷰 (Logical View)

- 목적: 시스템의 기능적 요구사항을 클래스, 객체, 상호작용으로 표현^{[1] [2]}.
- 대상: 설계자, 개발자^[3].
- 사용 다이어그램: 클래스 다이어그램, 상태 다이어그램, 시퀀스 다이어그램^{[1] [2]}.

2. 구현 뷰 (Implementation View)

- 목적: 개발 환경에서의 소프트웨어 모듈 구성과 의존성을 관리^{[1] [3]}.
- 대상: 개발자^[2].
- 사용 다이어그램: 컴포넌트 다이어그램, 패키지 다이어그램^{[1] [2]}.

3. 프로세스 뷰 (Process View)

- 목적: 병행 처리, 성능, 확장성 등 런타임 동적 특성을 분석^{[1] [3]}.
- 대상: 시스템 통합자^[2].
- 사용 다이어그램: 활동 다이어그램, 협력 다이어그램^{[1] [2]}.

4. 배포 뷰 (Deployment View)

- 목적: 물리적 하드웨어 환경에 컴포넌트 배치를 계획^{[1] [3]}.
- 대상: 시스템 엔지니어^[1].
- 사용 다이어그램: 배포 다이어그램^{[2] [3]}.

+1. 유스케이스 뷰 (Use Case View)

- 역할: 4개 뷰의 일관성을 검증하기 위한 시나리오 제공^{[2] [3]}.
- 적용 단계: 요구사항 분석 단계에서 주요 시나리오 선정^[4].
- 사용 다이어그램: 유스케이스 다이어그램^{[2] [3]}.

뷰 비교표

뷰 유형	주요 초점	이해관계자	대표 다이어그램
논리 뷰	기능 구현	설계자	클래스 다이어그램
구현 뷰	모듈 구성	개발자	컴포넌트 다이어그램
프로세스 뷰	시스템 성능	통합자	활동 다이어그램
배포 뷰	하드웨어 배치	시스템 엔지니어	배포 다이어그램
유스케이스 뷰(+1)	요구사항 검증	모든 이해관계자	유스케이스 다이어그램

이 모델은 1995년 Philippe Kruchten의 논문에서 처음 제안되었으며^[2], UML 표준과 결합되면서 발전해왔습니다^[4]. 각 뷰는 독립적으로 설계되지만 유스케이스 시나리오를 통해 상호 연계성을 검증함으로써 아키텍처의 완성도를 보장합니다^[4] [3].

✻

1. https://en.wikipedia.org/wiki/4+1_architectural_view_model
2. <https://moo-you.tistory.com/219>
3. <https://wandookong1.tistory.com/11>
4. <https://hsohe74.tistory.com/13>