

일반적인 소프트웨어 프로세스 모델들

■ 폭포수(waterfall) 모델

- 명세화와 개발이 독립적이고 구분되는 단계

■ 진화적 개발(Evolutionary 개발)

- 명세화와 개발이 섞여있음

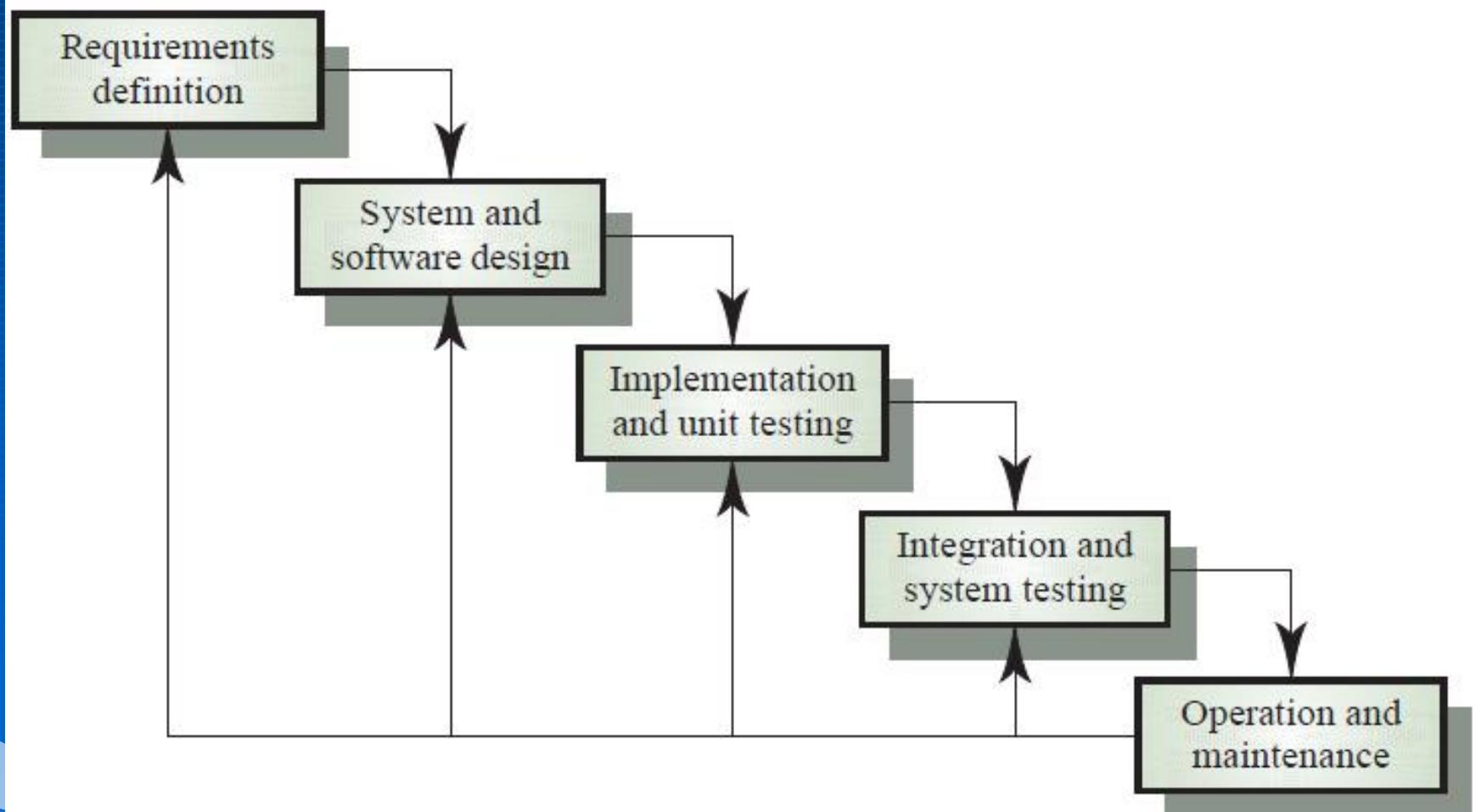
■ 정형적(Formal) 시스템 개발

- 수학적인 시스템 모델이 구현으로 공식적으로 변환됨.

■ 재사용 기반 개발

- 시스템이 기존의 컴포넌트들로부터 조립됨.

폭포수 모델



폭포수 모델 단계(phases)

- 요구사항 분석과 정의
- 시스템과 소프트웨어 설계
- 구현과 단위(unit) 테스팅
- 통합과 시스템 테스팅
- 작동과 유지보수(maintenance)
- 폭포수 모델의 단점은 프로세스가 지나간 후에 변경사항을 수용하기가 어렵다.

폭포수 모델 문제점

- 프로젝트를 개별적인 단계들로 분할하는 것의 비용통성
- 이것이 변경하는 고객의 요구사항에 응답하는 것을 어렵게 함.
- 그러므로, 이 모델은 요구사항이 잘 정의될 때에만 적절함.

진화적 개발

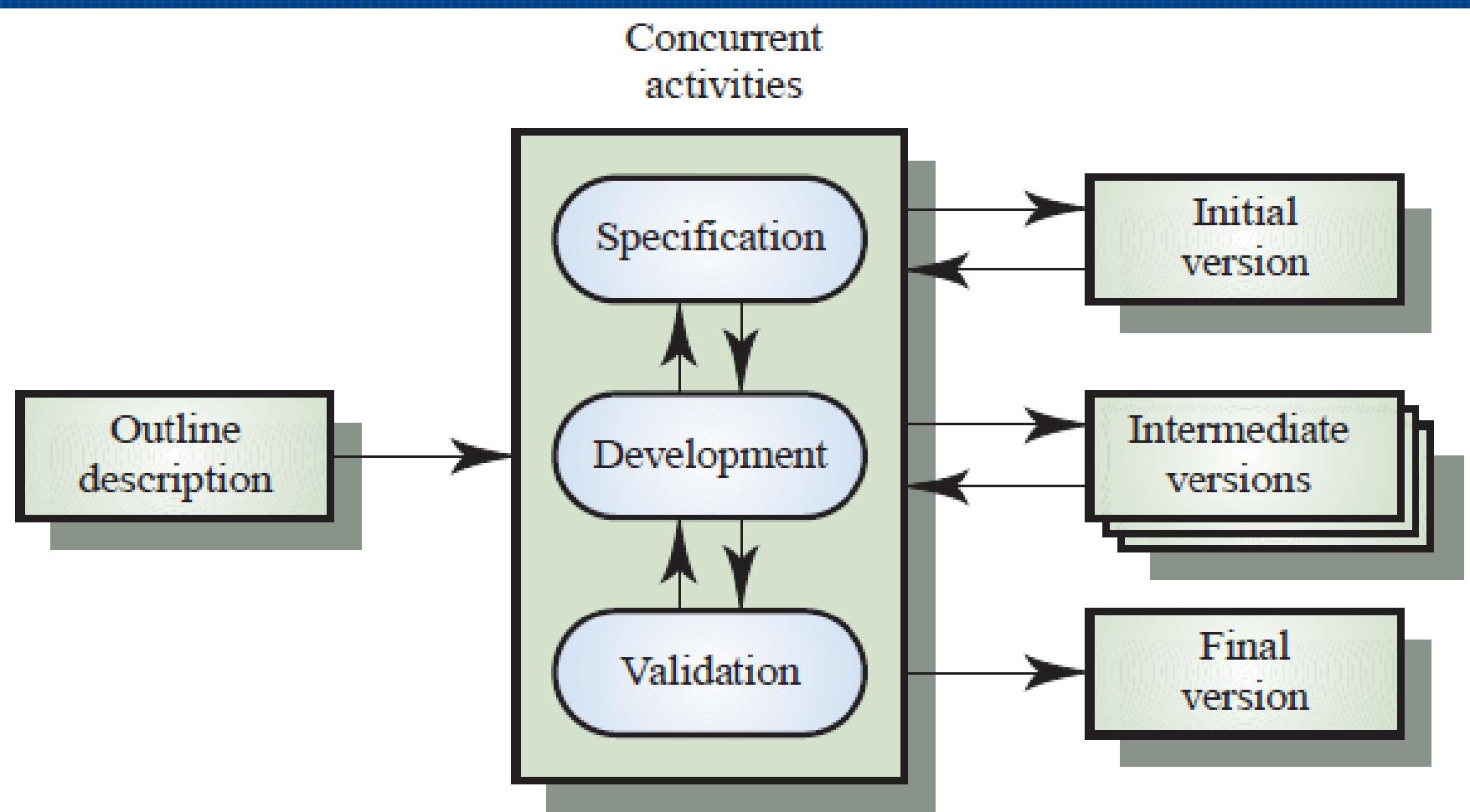
■ 탐험적(Exploratory) 개발

- 고객과 함께 작업하여 초기의 개략적 명세서로부터 최종 시스템을 개발하는 것이 목적. 요구사항이 잘 정의된 경우에 적합함.(상세한 명세서를 작성할 수 없을 때)

■ Throw-away 프로토타이핑

- 시스템 요구사항을 이해하는 것이 목적임. 요구사항을 잘 이해하지 못할 때 적합함.

진화적 개발(계속)



진화적 개발

■ 문제점

- 프로세스 가시성의 부족
- 종종 시스템의 구조가 나쁘게 됨.
- 특별한 기술(빠른 프로토타이핑을 위한 언어의 선택과 같은)이 필요할 수도 있음.

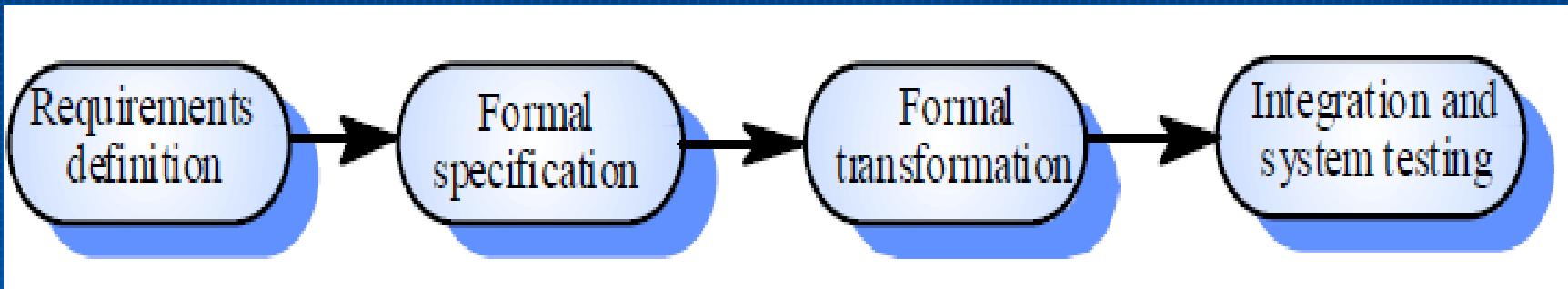
■ 적용성(Applicability)

- 작은 또는 중간 규모의 대화식 시스템
- 대형 시스템의 일부분(예, 사용자 인터페이스)
- 수명이 짧은 시스템

정형적 시스템 개발

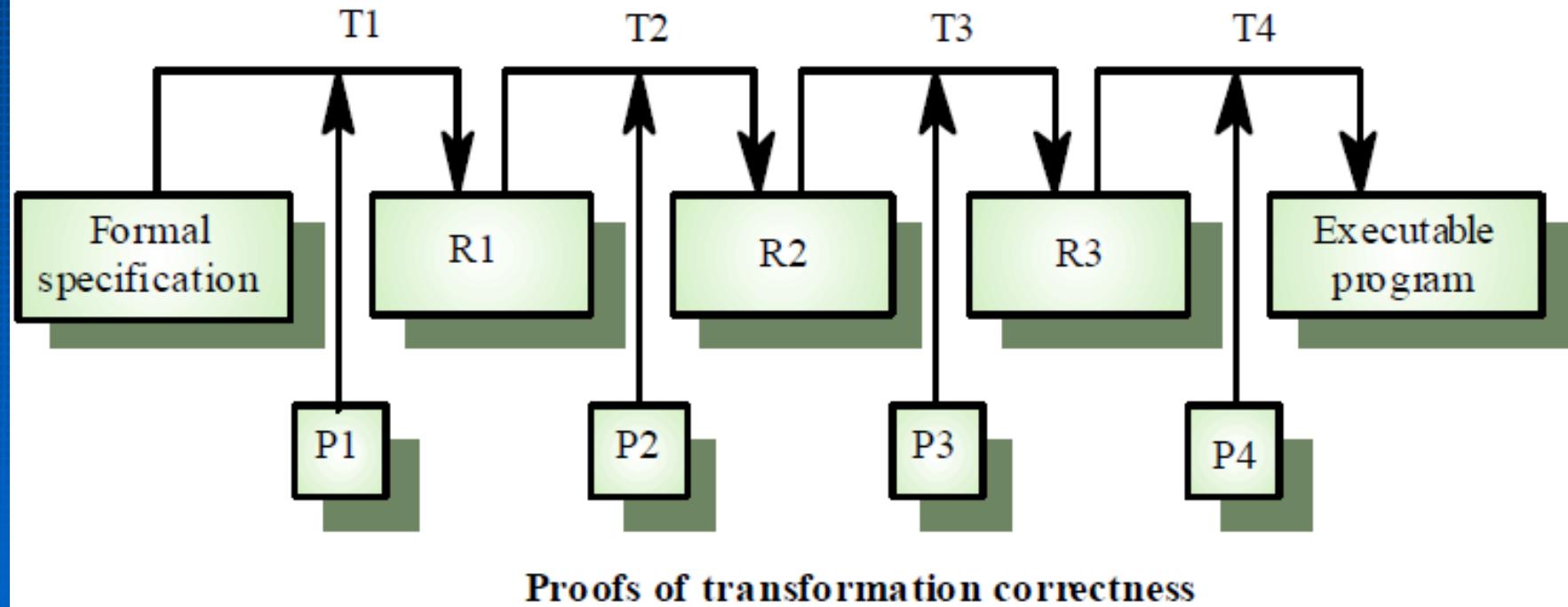
- 수학적 명세서로부터 다양한 표현방식을 거쳐 실행가능한 프로그램으로 변환하는 것에 기초
- 변환은 정확성을 유지함(correctness-preserving) 따라서 프로그램이 명세서에 적합하다는 것을 보이는 것은 매우 쉽다.
- 소프트웨어 개발의 청정실(Cleanroom) 접근법에 포함됨.

정형적 시스템 개발



정형적 변환

Formal transformations



정형적 시스템 개발

■ 문제점

- 전문적인 기술과 함께 기술을 적용하기 위한 교육이 필요.
- 시스템의 어떤 부분(예를 들면 사용자 인터페이스)들을 정형적으로 명세화하는 것이 어려움.

■ 적용성

- 시스템이 가동되기 전에 안전도 또는 보안이 확실하게 이루어져야하는 중요한(Critical) 시스템

재사용 중심 개발

- 시스템들이 기존의 컴포넌트들 또는 COTS (Commercial-off-the-shelf) 시스템으로부터 통합되는 것과 같은 체계적인 재사용에 기초
- 프로세스 단계
 - 컴포넌트 분석
 - 요구사항 수정
 - 재사용을 고려한 시스템 설계
 - 개발과 통합
- 이 접근법은 점점 중요해지고 있다. 그러나 아직도 이 방법에 대한 경험이 부족한 실정임.

재사용 중심 개발

