

정보처리기사

유튜브 주소 : <https://youtu.be/lwgjhEE21RQ>

소프트웨어 생명주기 모델

소프트웨어 개발 방법론

요구사항 확인

소프트웨어 생명주기 모델

- 시스템의 요구분석부터 유지보수까지 전 공정을 체계화한 절차
- 생명주기 모델 프로세스
 - 요구사항 분석
 - 요구사항을 고려하여 제품에 부합하는 요구와 조건을 결정
 - 설계
 - 시스템 명세 단계에서 정의한 기능을 수행할 수 있도록 방법을 논리적으로 결정
 - 시스템 구조, 프로그램, 사용자 인터페이스 설계
 - 구현
 - 설계 단계에서 결정한 방법을 특정 프로그래밍 언어를 사용해 실제 프로그램을 작성
 - 인터페이스 개발, 자료 구조 개발, 오류 처리 개발
 - 테스트
 - 시스템이 요구를 만족하는지, 예상과 실제 결과가 어떤 차이를 보이는지 검사 및 평가
 - 단위 테스트, 통합 테스트, 시스템 테스트, 인수 테스트
 - 유지보수
 - 시스템이 인수되고 설치된 후 일어나는 모든 활동

소프트웨어 생명주기 모델

- 소프트웨어 생명주기 모델 종류
 - 폭포수 모델
 - 각 단계를 확실히 마무리 지은 후 다음 단계로 넘어가는 모델
 - 프로토타이핑 모델
 - 주요 기능을 프로토타입으로 구현한 후 피드백을 반영하며 만들어가는 모델
 - 나선형 모델
 - 점진적으로 완벽한 시스템으로 개발해나가는 모델
 - 반복적 모델
 - 구현 대상을 나누어 병렬적, 혹은 반복적으로 개발하여 점증 완성시키는 모델

소프트웨어 생명주기 모델

- 폭포수 모델

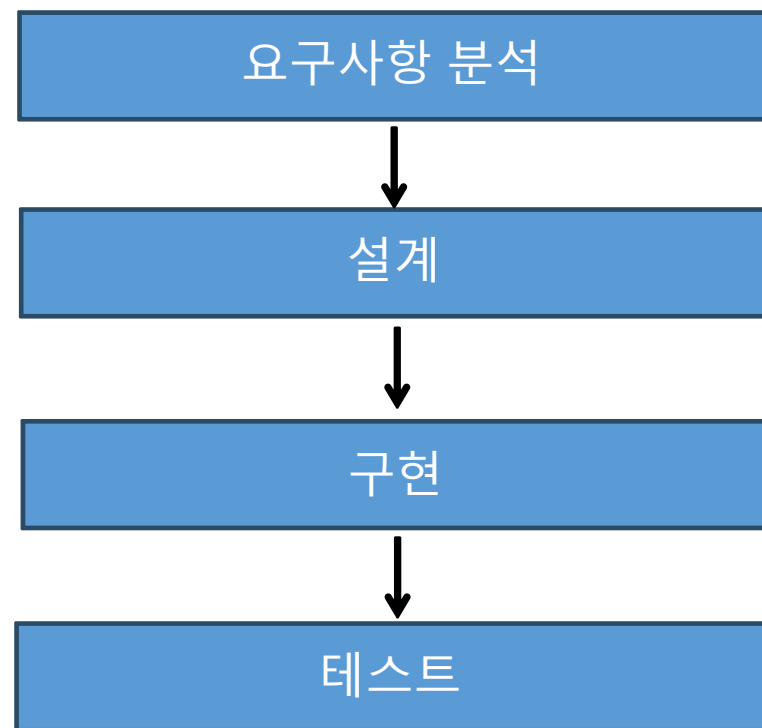
- 각 단계를 확실히 마무리 지은 후 다음 단계로 넘어가는 모델

- 가장 오래된 모델

- 모형의 적용 경험과 성공 사례가 많음

- 장점 : 단계별 정의와 산출물이 명확

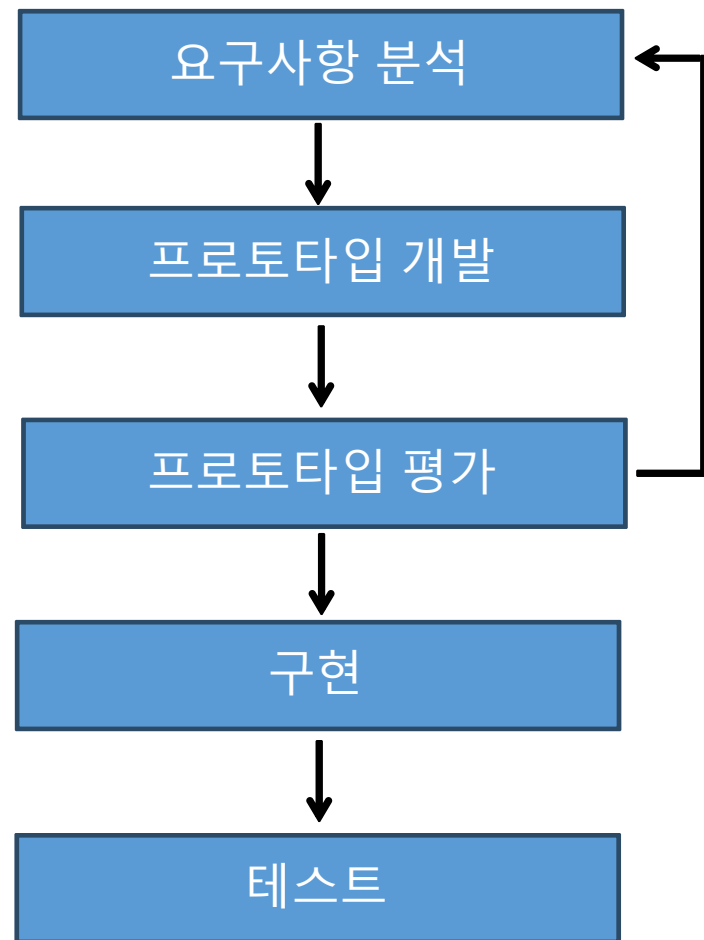
- 단점 : 요구사항 변경이 어려움



소프트웨어 생명주기 모델

- 프로토타이핑 모델

- 주요 기능을 프로토타입으로 구현한 후 피드백을 반영하며 만들어가는 모델
- 발주자, 개발자 모두에게 공동의 참조 모델 제공
- 프로토타입을 구현 단계의 골격으로 사용 가능
- 장점 : 요구분석 용이
- 단점 : 프로토타입 폐기에 따른 비용 증가



소프트웨어 생명주기 모델

- 나선형 모델

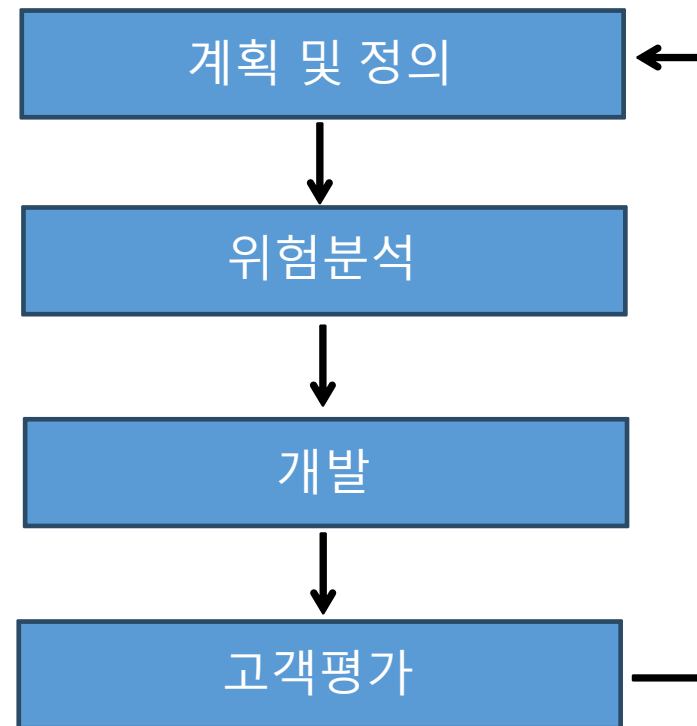
- 점진적으로 완벽한 시스템으로 개발해나가는 모델

- 위험 분석, 반복 개발

- 변경에 유연한 대처 가능

- 장점 : 위험성 감소

- 단점 : 단계 반복에 따른 관리 어려움



소프트웨어 생명주기 모델

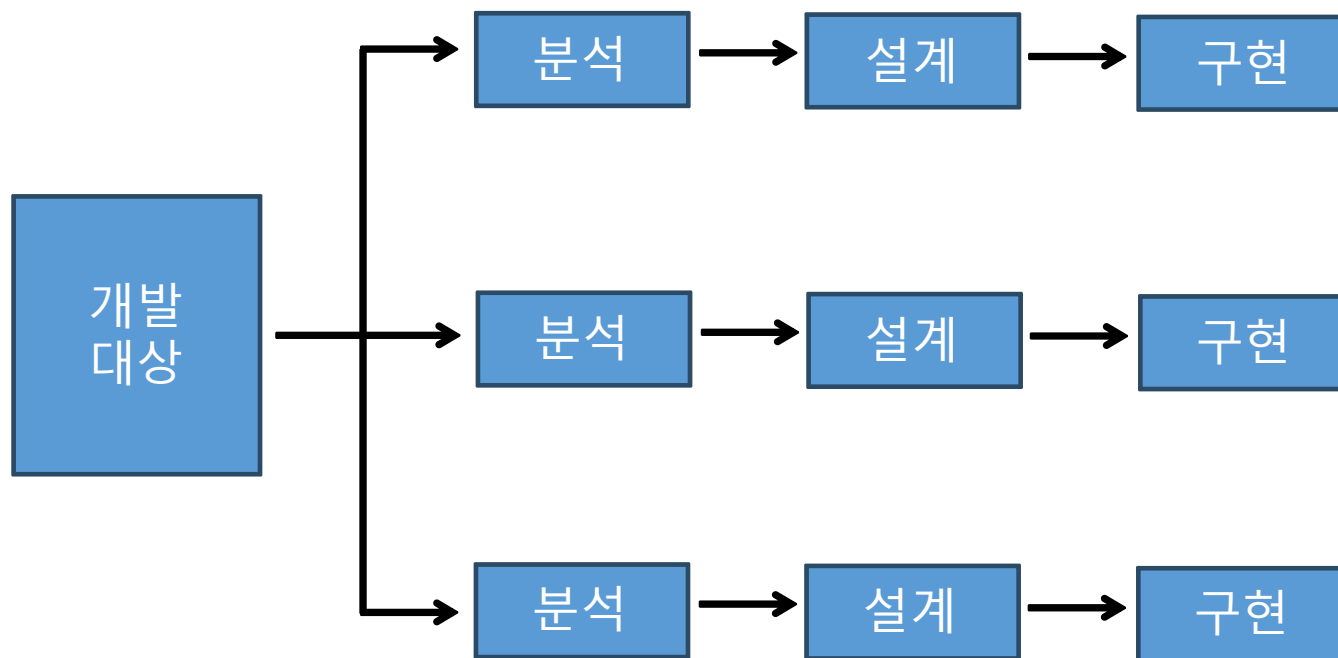
- 반복적 모델

- 구현 대상을 나누어 병렬적, 혹은 반복적으로 개발하여 점증 완성시키는 모델

- 증분방식으로 병행 개발

- 장점 : 일정 단축 가능

- 단점 : 관리 비용 증가



소프트웨어 개발 방법론

- 개발 전 과정에 지속적으로 적용할 수 있는 방법, 절차, 기법
- 개발 방법론 종류
 - 구조적 방법론
 - 기능에 따라 나누어 개발하고 통합하는 분할과 정복 접근 방식의 방법론
 - 정보공학 방법론
 - 개발에 필요한 관리 절차와 작업 기법을 체계화한 방법론
 - 객체 지향 방법론
 - 객체 단위로 시스템을 분석 및 설계하는 방법론
 - 컴포넌트 기반 방법론
 - 컴포넌트를 조립해서 하나의 새로운 응용 프로그램을 완성하는 방법론
 - 애자일 방법론
 - 변화에 유연하고 신속하게 적응하면서 효율적으로 시스템을 개발할 수 있는 방법론
 - 제품 계열 방법론
 - 특정 제품에 적용하고 싶은 공통된 기능을 정의하여 개발하는 방법론

소프트웨어 개발 방법론

- 애자일 방법론

- 기존 개발 방법론의 한계를 극복하기 위해 등장

등장 배경	설명
소프트웨어 개발 환경의 변화	소프트웨어 개발 트렌드가 모바일 환경으로 변화 시장 적시성과 잦은 배포의 중요성 부각
기존 개발 방법론의 한계	전통적 방법론은 문서 및 절차 위주 -> 변화에 신속한 대응 어려움 빠르게 적용하고 효율적으로 개발할 수 있는 방법론의 필요성 증가

- 절차보다는 사람이 중심
- 신속 적응적 경량 개발 방법론
- 개발 기간이 짧고 신속



소프트웨어 개발 방법론

- 애자일 방법론과 전통적 방법론 비교

비교 대상	전통적 방법론	애자일 방법론
계획수립	확정적 범위 설정	유동적 범위 설정
업무수행	관리자 주도적 명령과 통제 , 개인 단위 업무	팀 중심 업무 수행
개발 / 검증	분석 / 설계 / 구현 / 테스트를 순차적 수행	반복 주기 단위로 소프트웨어를 개발 / 검증
팀관리	경쟁, 개별 평가	업무 몰입, 팀 평가
문서화	상세한 문서화를 강조	문서화보다는 코드를 강조
성공요소	계획 / 일정 준수	고객 가치 전달
유형	폭포수, 프로토타입, 나선형	XP, 스크럼, 린

소프트웨어 개발 방법론

- 애자일 방법론 유형
- XP(eXtreme Programming)
 - 의사소통 개선과 즉각적 피드백으로 소프트웨어 품질을 높이기 위한 방법론
 - 1~3주의 반복 개발 주기
- 스크럼(SCRUM)
 - 매일 정해진 시간, 장소에서 짧은 시간의 개발을 하는 팀을 위한 방법론
 - 프로젝트 관리 중심 방법론
- 린(LEAN)
 - 낭비 요소를 제거하여 품질을 향상시킨 방법론
 - 도요타의 린 시스템 품질 기법을 소프트웨어 개발 프로세스에 적용

요구사항 확인

- 요구 공학
 - 사용자의 요구가 반영된 시스템을 개발하기 위함
 - 요구사항에 대한 도출, 분석, 명세, 확인 및 검증을 하는 구조화된 활동
- 요구 공학 목적
 - 이해 관계자 사이에 효과적인 의사소통 수단 제공
 - 시스템 개발의 요구사항에 대한 공통된 이해 설정
 - 요구사항 누락 방지 및 이해 오류로 인한 불필요한 비용 절감
 - 초기 요구사항 관리로 개발 비용과 시간 절약

요구사항 확인

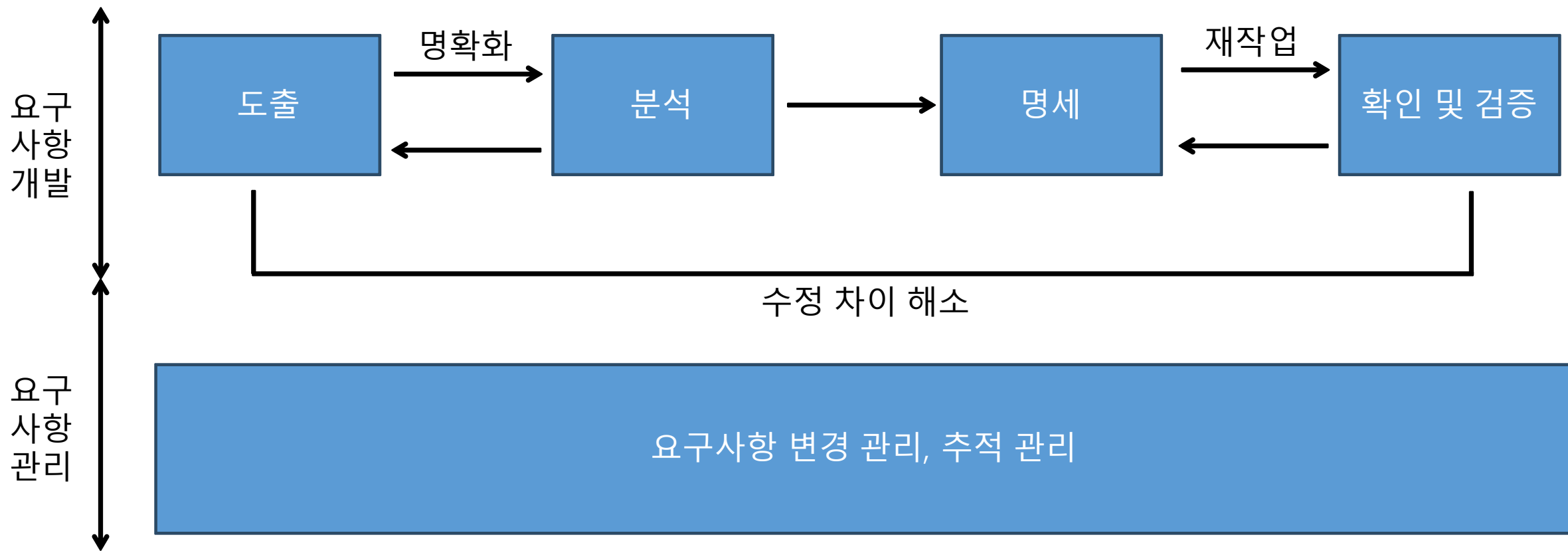
• 요구사항 분류

구분	기능적 요구사항	비기능적 요구사항
개념	시스템이 제공하는 기능, 서비스에 대한 요구사항	시스템이 수행하는 기능 이외의 사항
도출 방법	특정 입력에 대한 시스템의 반응 특정 상황에 대한 시스템의 동작	품질 속성에 관련하여 시스템이 갖춰야 할 기술 시스템이 준수해야 할 제한 조건에 관한 기술
특성	기능성, 완전성, 일관성	신뢰성, 사용성, 유지보수성, 이식성, 보안성
사례	쇼핑몰 홈페이지에서의 장바구니 기능 제공 상품 결제 -> 신용카드 결제 가능해야함	특정 함수 호출 시간 3초 이내 시스템은 24시간 내내 가동되어야 함 가동률 99.5%를 만족해야 함

요구사항 확인

- 요구공학 프로세스

- 요구사항 개발 단계 & 요구사항 관리 단계로 구성



요구사항 확인

- 요구사항 도출

- 소프트웨어가 해결해야 할 문제 이해
- 고객으로부터 제시되는 요구에 대한 관련 정보 식별 후 수집 방법 결정
- 수집된 요구 사항을 구체적으로 표현
- 이해관계자와 효율적인 의사소통이 중요
- 고객 분석, 조직 환경 분석, 후보 요구사항 분류, 요구사항 소스 관리

요구사항 확인

- 요구사항 분석
 - 요구사항에 대해 충돌, 중복, 누락 등의 분석을 통해 완전성과 일관성 확보
 - 요구사항들 간 상충되는 것들을 해결
 - 소프트웨어의 범위 파악
 - 소프트웨어의 환경 간 상호 작용 원리 이해
 - 시스템 요구사항 정제, 분류, 모델링, 우선순위 부여 등 역할 수행
 - 요구사항 정의 문서화 수행

요구사항 확인

- 요구사항 명세
 - 체계적으로 검토, 평가, 승인될 수 있는 문서를 작성하는 단계
 - 동의한 요구사항을 하나 이상의 형태로 저장
 - 정형화된 요구사항을 생성
 - 요구사항 명세 기준 정의, 명세서 작성, 요구사항 추적 관련 정보 저장 등 수행

요구사항 확인

- 요구사항 확인 및 검증
 - 분석가가 요구사항을 이해했는지 확인
 - 요구사항 문서가 회사의 표준에 적합한지 검증
 - 요구사항 정의 문서들에 대한 형상 관리 수행
 - 요구사항 베이스 라인 수립
 - 요구사항 용어 검증

Q & A