

폭포수 모델 - 단계별 산출물 모델

진행과정: 요구사항 분석->설계->구현->테스트->유지보수

원형 모델

진행과정: 요구사항 분석->시제품 설계->시제품 개발->고객의 시제품 평가->시제품 정제->완제품 생산

가장 중요하게 평가되는 단계는 고객의 평가 단계

장점-고객의 요구사항 초기 도출

나선형 모델 - 계획 및 정의 단계, **위험 분석 단계**, 개발 단계, 고객 평가 단계

4세대 기법

특징

- CASE를 비롯한 자동화 도구들을 이용 -> 요구사항 명세서로부터 실행 코드를 자동 생성 방법
- 사용하는 고급 언어 수준에서 요구사항이 명시 -> 실행될 수 있는 제품으로의 전환 가능
- 고급 언어를 실행 코드로 바꾸어 줄 만큼 정교하지 못하여 고급 언어의 모호성을 해결하기 위해서 정형 명세 언어로 표현하려는 노력이 진행
- 성능이 뛰어나지 못하여 유지보수가 어렵다.

객체지향 기법

애자일

가치

프로세스와 도구<개인과 협업 / 포괄적인 문서화<제대로 작동하는 소프트웨어 / 계약 협상<고객과의 협력
계획 < 변화에 대응

기본-단순성, 의사소통, 용기, 자신감, 자발적 헌신, 존중, 피드백

원리-점증적인 인도, 프로세스가 아닌 사람 중심, 상호 협력

특징

- 지속적인 개발 촉진
- 설계가 잘 되도록 지속적인 개선
- 빠른 인도 주기
- 개발 후반부의 요구사항 변화도 긍정적 수용

관리

- 팀 주도적 리더십과 협업
- 요구사항의 지속적인 상세화
- 코딩과 테스트 동시적 반복 수행
- 빠른 릴리즈
- 고객과의 지속적인 의사소통
- 초기화 이후 잦은 통합

어려움

- 아직 많은 사람들에게 낯선 프로세스
- 고객 요구되는 높은 참여도와 이해도
- 구성원들 요구하는 소프트웨어 개발 프로세스 이해와 객체지향 기술

종류-XP, SCRUM, Kanban, Lean Process

XP(익스트림 프로그래밍)-고객에게 최고의 가치를 가장 빨리

가치: 의사소통, 단순함, 피드백, 용기, 존중

특징-고객과 팀 중심, 빠른 개발 속도, 사용자 스토리, 최소화, 단위 테스트 생성, 시험 우선 개발, 짝 프로그래밍, 자동화된 시험 장치 사용

고객의 책임-사용자 스토리 작성, 테스트 명세서 작성, 테스트 실행

짝 프로그래밍 특징-공동 책임을 지는 환경 조성, 비정형적 코드 검사, 리팩토링 지원

리팩토링

특징

-기존 코드의 설계를 개선 기술

-기술적 부채->점진적으로 갚아 나가는 좋은 방법

-회귀 시험 필요

-적은 비용으로 수정

사용자 스토리

구성-대화, 카드, 확인

특징

-반복적 개발에 효과적으로 사용

-문서보다 구두 의사소통 강조

-고객이나 개발자 모두 이해 가능

컴포넌트 기반(조립)

특징

-조립->원하는 소프트웨어 신속 개발

-잘 만들어진 부품을 사다가 조립하여 소프트웨어 개발

-세부적인 구현된 내부의 구현 사항들->외부 감추고 외부적인 인터페이스만 제공

-하나 이상의 프로그램들을 하나의 단위로 관리하는 패키지

-동적 바인드->실행 시간 인터페이스 접근 가능

-조립하여 프로그램 구성->Plug-In-Play식의 개발 가능

장점

-컴포넌트 단위 분할해서 쉽게 관리

-높은 품질의 소프트웨어를 얻음(품질 향상)

-재사용성이 높고 개발 시간을 단축

-느슨한 결합도와 큰 입자의 특징

-내부 구현 관여X->서로의 계약을 통한 조합

-구현 언어 구매 받지X->상호 간 호환성 인터페이스 연동 가능

-개발 비용 절감 및 생산성 향상

UML 매서드 종류와 기호

Public	+
Protected	#
Private = Default	-