



CS 스터디(Agile)

소프트웨어 공학

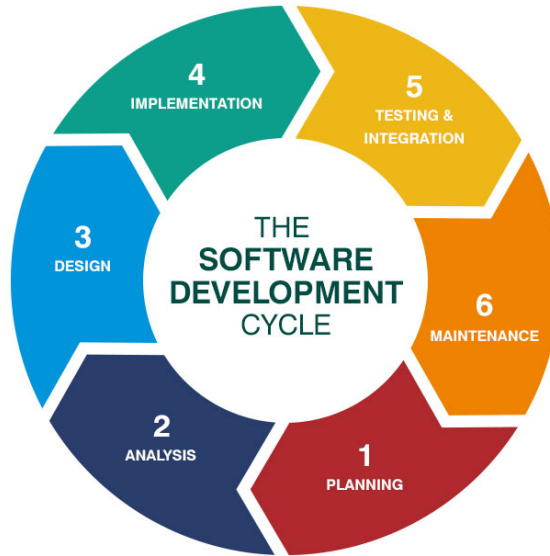
- 소프트웨어의 개발, 운용, 유지보수 등 생명 주기 전반을 체계적이고 서술적이며 정량적으로 다루는 학문

소프트웨어 공학의 세부 분야

- 소프트웨어 요구사항
- 소프트웨어 설계
- 소프트웨어 개발
- 소프트웨어 시험
- 소프트웨어 유지 보수
- 소프트웨어 형상 관리
- 소프트웨어 공학 관리
- 소프트웨어 개발 프로세스
- 소프트웨어 공학 도구
- 소프트웨어 품질

소프트웨어 개발 프로세스

소프트웨어 제품을 개발하기 위해 필요한 과정 또는 구조 (=소프트웨어 생명주기)



<https://velog.io/@dubu4050/Testing>

- 개발 모형
 - 애자일
 - 나선 모형
 - 폭포수 모델
 - TDD

Agile software development

No 계획 → 예측 불가, 효율이 낮음

Too many 계획 → 전체적인 개발 흐름 자체를 느리게 함

- 개발 방법 사이에서 타협점을 찾고자 하는 방법론

전통적인 개발기법들(폭포수)의 단점은 지나치게 계획과 절차에 의존

→ 고객의 요구사항은 계속해서 바뀜

→ 전통적인 개발 기법에 적용하기 어려움

→ 애자일의 등장 배경

→ 빠르고 낭비없이 변화에 기민하게(Agile) 대응 하는 개발을 가능하게 하는 다양한 방법론 전체를 일컫는다!

→ 특정 개발 방법론 X

주요 특성

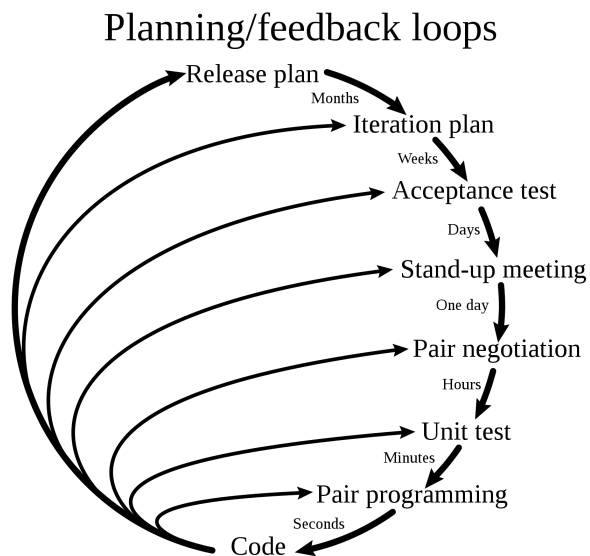
- 개인과 개인 간의 상호작용이 프로세스 보다 우선
- 작동하는 소프트웨어가 포괄적인 문서보다 우선
- 고객과의 협업이 계약 협상보다 우선
- 변화에 대응하는 것이 계획을 따르는 것보다 우선

종류

- 익스트림 프로그래밍(Extreme Programming, XP)

비즈니스 상의 요구가 시시각각 변동이 심한 경우에 적합한 개발 방법

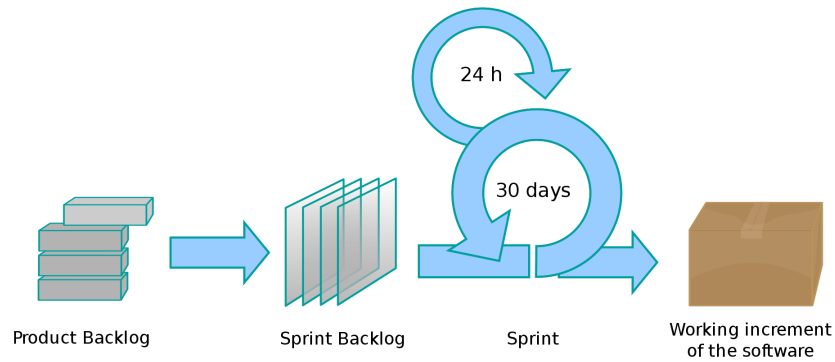
→ 애자일 개발 프로세스의 대표적인 개발 방법 중 하나



https://ko.wikipedia.org/wiki/익스트림_프로그래밍#/media/파일:Extreme_Programming.svg

- 스크럼

30일 마다 동작 가능한 제품을 제공하는 스프린트(Sprint)를 중심으로 진행. 매일 정해진 시간, 정해진 장소에서 짧은 시간의 개발을 하는 팀을 위한, 프로젝트 관리 중심의 방법론



- 제품 백로그(Product Backlog)
개발할 제품에 대한 요구 사항 목록
- 스프린트(Sprint)
반복적인 개발 주기
- 스프린트 백로그(Sprint Backlog)
각각의 스프린트 목표에 도달하기 위해 필요한 작업 목록
- 크리스털 패밀리
- Feature-Driven Development
- Adaptive Software Development
- 익스트림 모델링