

애자일

소프트웨어 방법론의 하나로, 처음부터 끝까지 계획을 수립하고 개발하는 **폭포수**(Waterfall) 방법론과는 달리 개발과 함께 즉시 피드백을 받아서 유동적으로 개발하는 방법이다.

정식 명칭은 애자일 소프트웨어 개발(Agile Software Development). 한국에서는 주로 **애자일 방법론** 이라고 부른다. 켄트 벡이 주창한 익스트림 프로그래밍(XP, Extreme Programming)과 테스트 주도 개발이 대표적이다.

애자일 선언문(Manifesto for Agile Software Development)

우리는 소프트웨어를 개발하는 더 나은 방법을 발견하기 위해 일하며, 다른 사람들이 그것을 할 수 있도록 돕는다. 이 작업을 통해 우리는 다음을 중요시한다는 것을 깨달았다:

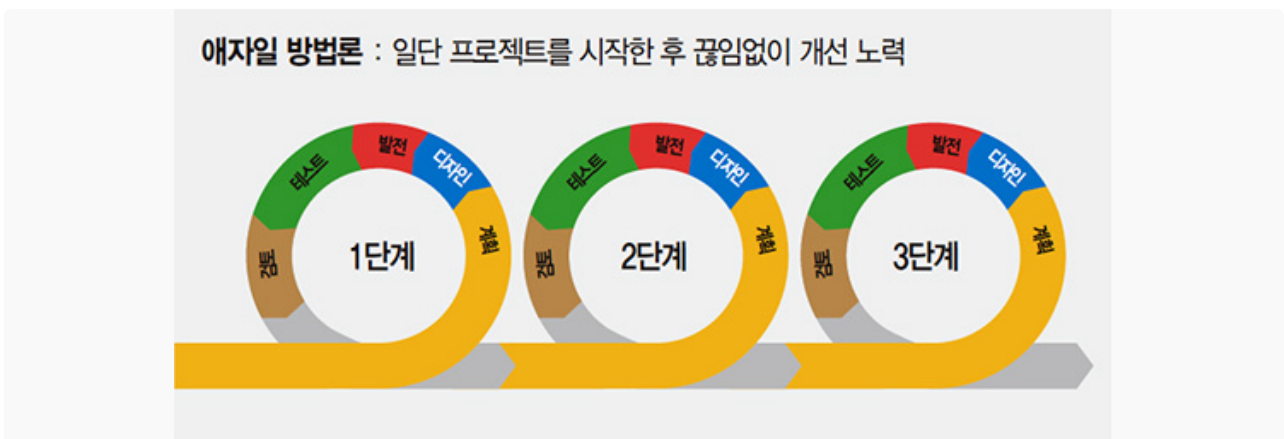
- 과정과 도구보다 **개인**과 상호작용
- 포괄적인 문서보다 **작동하는 소프트웨어**
- 계약 협상보다 **고객과의 협력**
- 계획에 따르기보다 **변화에 대응하기**

가치 있게 여긴다.

이 말은, 왼쪽에 있는 것들도 가치가 있지만, 우리는 **오른쪽에 있는 것들에 더 높은 가치를 둔다** 는 것이다.

애자일 방법론의 진행 과정

애자일 방법론은 **계획 → 설계(디자인) → 개발(발전) → 테스트 → 검토(피드백)** 순으로 반복적으로 진행된다.



- **계획 및 분석** : 고객과 사용자가 원하는 바를 파악하여 타당성을 조사하고 SW 기능과 제약조건을 정의하는 명세서 작성, 대상이 되는 문제 영역과 사용자가 원하는 task를 이해하는 단계
- **설계(디자인)** : 기획 의도에 맞는 설계 및 디자인 추가 및 수정하는 단계
- **개발(발전)** : 설계단계에서 만들어진 설계서를 바탕으로 프로그램을 작성, 코딩, 디버깅, 단위/통합테스트 수행

- **테스트** : 발생할 수 있는 실행 프로그램 오류를 발견, 수정하는 단계
- **검토(피드백)** : 기획 의도를 파악하고 시험 결과와 기획에 따라 수정할 부분을 제시하는 단계

애자일 방법론의 특징

- 고객과 개발자의 지속적인 소통을 통하여 변화하는 요구사항을 신속하게 수용한다.
- 개발자 개인의 가치보다는 팀의 목적을 우선시하며 고객의 의견을 가장 우선시 한다.
- 팀원들과의 주기적인 회의 및 제품 시현을 통한 방지를 점검한다.
- 진행하면서 프로그램을 시행해보고 고객으로부터 피드백을 받는다.
- 내부 구조 형성을 통한 비용 절감에 힘쓰는 동시에 프로그램 품질 향상을 위해 노력한다.

애자일 방법론의 장단점

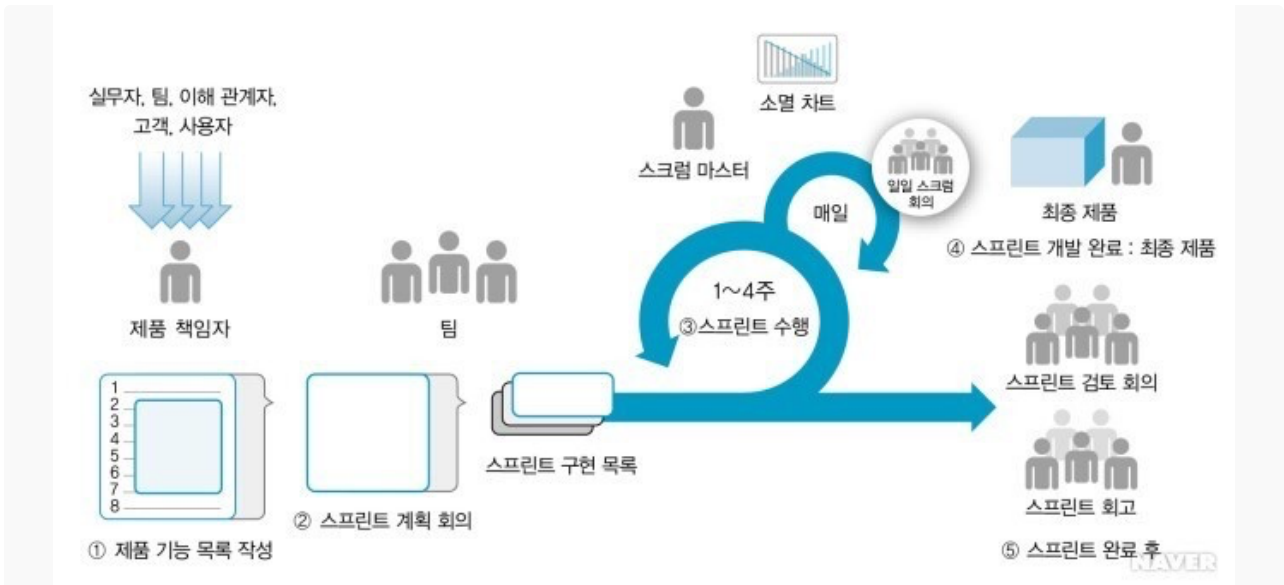
장점

- 프로젝트 계획에 걸리는 시간을 최소화할 수 있다.
- 점진적으로 테스트 할 수 있어서 버그를 쉽고 빠르게 발견할 수 있다.
- 계획 혹은 기능에 대한 수정과 변경에 유연하다.
- 고객 요구사항에 대한 즉각적인 피드백에 유연하며 프로토타입 모델을 빠르게 출시 할 수 있다.
- 빠른 기한의 프로젝트를 빠르게 출시할 수 있다.

단점

- 확정되지 않은 계획 및 요구사항으로 인한 반복적인 유지보수 작업이 많다.
- 고객의 요구사항 및 계획이 크게 변경되면 모델이 무너질 수 있다.
- 개인이 아닌 팀이 중심이 되다 보니 공통으로 해야 할 작업이 많을 수 있다. (회의, 로그 등)
- 반복적인 업무로 속도는 빠를 수 있으나 미흡한 기능들에 대한 대처가 필요하다.
- 확정되지 않은 계획으로 개발 진행 시 이해하지 못하고 진행하는 부분이 많을 수 있다.

스크럼(Scrum)



스크럼은 애자일 프레임워크 중 하나로 반복적이고 점진적인 개발방법을 말하며, '스프린트'라고 불리는 작업 단위를 사용하여 작업을 추정하는 프로젝트 계획 방법입니다.

스크럼 진행과정

1. 제품 백로그(product backlog) 작성

제품 백로그는 이해관계자로부터 추출된 제품이 제공해야 하는 기능이나 개발할 제품에 대한 요구사항 목록을 말합니다.

2. 스프린트 계획 회의

제품 책임자, 스크럼 마스터, 개발팀이 참여하며 제품 백로그를 기반으로 스프린트 목표와 스프린트 백로그를 계획하는 회의

3. 스프린트 백로그(sprint backlog) 작성

제품백로그에서 결정된 우선순위를 기반으로 스프린트 동안 해야 하는 일에 대한 리스트를 스프린트 백로그라고 합니다. 스프린트 목표를 구현 가능하도록 각각의 요구사항을 task 단위로 나누어 개발자들이 나눠서 작업을 수행하며, 업무 보드 형식의 스프린트 백로그를 만드는데 이것을 **스크럼 보드(Scrum Board)** 라고 한다.

4. 일일스크럼 미팅(daily scrum meeting)

어제 했던일과 오늘 할일, 수행중 문제점이나 장애요인 등을 공유하며 문제가 있을 경우 미팅 후 바로 해결하며, 일일 스크럼 미팅을 함으로서 프로젝트 후반부에 문제점이 갑자기 발생하는 것을 예방합니다.

5. 실행 가능한 제품 개발

각 스프린트마다 목표는 실행 가능한 제품을 만들어서 배포하는 것이며, 매 스프린트의 결과로써 나오는 산출물을 잠재적으로 출시 가능한 제품 증분(Potentially Shippable Product Increment)이라고 부릅니다.

6. 스프린트 리뷰

스프린트가 종료되었을 때, 개발팀이 스프린트동안 개발한 증분의 기능을 고객을 포함한 이해관리자들에게 보여주고 피드백을 받는 과정을 말합니다.

7. 스프린트 회고(sprint retrospective)

스프린트 리뷰 후 프로젝트를 진행하면서 좋았던점이나 문제점, 미진한 점 등을 도출함으로써 다음 스프린트를 보다 더 나은 방향으로 개선할 수 있도록 하는 과정을 말합니다.

8. 다음 스프린트 시작

스프린트는 제품 책임자가 제품을 출시할 준비가 되었다고 판단할 때까지 계속되며, 스프린트가 회고 후 휴식 기간없이 다음 스프린트를 진행합니다.

스크럼의 장단점

장점

- 스프린트마다 생산되는 실행 가능한 제품을 통해 사용자와 의견을 나눌 수 있음
- 회의를 통해 팀원들간 신속한 협조와 조율이 가능
- 자신의 일정을 직접 발표함으로써 업무 집중 환경 조성
- 프로젝트 진행 현황을 통해 신속하게 목표와 결과 추정이 가능하며 변화 시도가 용이함

단점

- 추가 작업 시간이 필요함 (스프린트마다 테스트 제품을 만들어야하기 때문)
- 15분이라는 회의 시간을 지키기 힘들 (시간이 초과되면 그만큼 작업 시간이 줄어듦)
- 스크럼은 프로젝트 관리에 무게중심을 두기 때문에 프로세스 품질 평가에는 미약함