

3주차 주제 TDD (테스트 주도 개발)

TDD가 무엇인가

테스트 주도 개발 시작하기

TEST-DRIVEN DEVELOPMENT
작동하는 검증한 코드를 만드는 데 필요한 습관

최범균 지

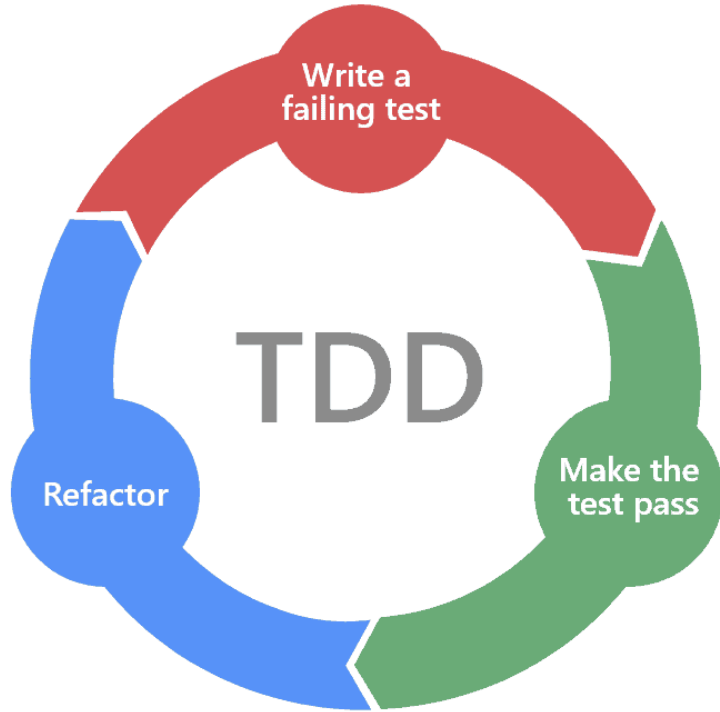


KM 가메출판사

TDD란?

- Test Driven Development의 약자. 우리말로 테스트 주도 개발
- 반복 테스트를 이용한 소프트웨어 방법론으로 작은 단위의 테스트 케이스를 작성하고 이를 통과하는 코드를 추가하는 단계를 반복하여 구현한다.
- 이 기법을 개발했거나 ‘재발견’한 것으로 인정되는 켄트 벡(Kent Beck)은 TDD가 단순한 설계를 장려하고 자신감을 불어넣어 준다고 말하였다.
- 최근에는 테스트 코드가 동작을 테스트하기 위해 사용될 뿐 아니라, ci 도구 사용 시 테스트 코드의 성공 여부를 확인해 전부 성공한 경우에만 pr을 통과시키거나 운영브랜치로 merge시키는 등의 추가 동작을 하는데도 많이 사용된다고 한다.

TDD 순서



1단계 ●) 기능을 검증하는 테스트 코드 작성

2단계 ●) 테스트를 통과하는 기능 개발

3단계 ●) 작성한 코드 리팩토링

- TDD는 기본적으로 3단계의 반복으로 진행하며 점진적으로 코드를 개선해나가며 개발이 진행된다.
- 왜 1단계가 Write a failing test (실패한 테스트 작성)일까?

기능을 개발하기 전, 테스트 코드를 우선으로 작성하기 때문에 당연히 그 테스트는 실패하는 테스트일 것이다.

- 2단계에서 이 실패하는 테스트를 성공시키기 위한 기능 개발을 하면 된다.

간단한 TDD 예제

https://velog.io/@chae_ag/TDD.-%ED%85%8C%EC%8A%A4%ED%8A%B8-%EC%A3%BC%EB%8F%84-%EA%B0%9C%EB%B0%9C%EC%9D%B4-%EB%AC%B4%EC%97%87%EC%9D%B8%EC%A7%80-%EC%95%8C%EC%9E%90

TDD의 장점

1. 기능 단위로 테스트하기 때문에 문제의 원인을 알기 쉽다.

- 보통 개발을 다 해놓고 나서 테스트를 하는 ATDD(인수테스트 주도 개발)를 하는데, 테스트 중 혹여나 문제를 발견한다면 정확하게 문제의 원인이 무엇인지 진단하기 힘들다.
- 하지만 TDD를 사용하면 기능 단위로 테스트하기 때문에 문제의 원인을 알기 쉽다.

2. 변화에 대한 두려움 해소

- 테스트 코드를 먼저 작성하고 기능 개발을 하기 때문에 개발한 기능이 잘 동작할지에 대한 불안감을 없앨 수 있다.

TDD의 장점

3. 프로그래머의 오버 엔지니어링을 방지한다.

- 개발을 하다 보면 간혹 계획하지 않았던 코드를 추가하여 오버 엔지니어링하는 경우가 있다. 하지만 TDD의 원칙 중 하나는, 테스트를 통과하기 위한 최소한의 코드만 작성 및 개선해야 한다는 것이다.
- 기능 단위로 테스트를 진행하기 때문에, 문제가 발견되지 않은 코드에 영향을 줄 수 있는 오버 코딩은 하지 않는다.

TDD의 단점

1. TDD를 익히는 데 많은 시간이 걸린다.

TDD는 마치 운동과 같다.

운동을 꾸준히 하면 건강해지고 체력이 좋아지는 것처럼

TDD도 꾸준히 연습하고 적용해야 실력이 늘고 효과를 볼 수 있다.

-저자 최범균 [테스트 주도 개발 시작하기] 255p

2. 생산성 저하

- TDD에 대한 프로그래머들의 의견은 늘 엇갈리는데, 그 이유 중 가장 큰 것이 이 생산성 저하 문제 때문일 것이다.

TDD를 지키면서 개발하려면 처음부터 2개의 코드를 짜야 하고, 중간중간 테스트를 하면서 고쳐나가야 하기 때문에 개발 속도가 느려지기 때문에 생산성이 저하된다.

끝

- TDD가 무조건 좋은 것은 아니다...
- 단위 테스트를 제대로 하자...

도움 받은 곳

<https://jwdeveloper.tistory.com/280>

<https://jaehoney.tistory.com/283>

