

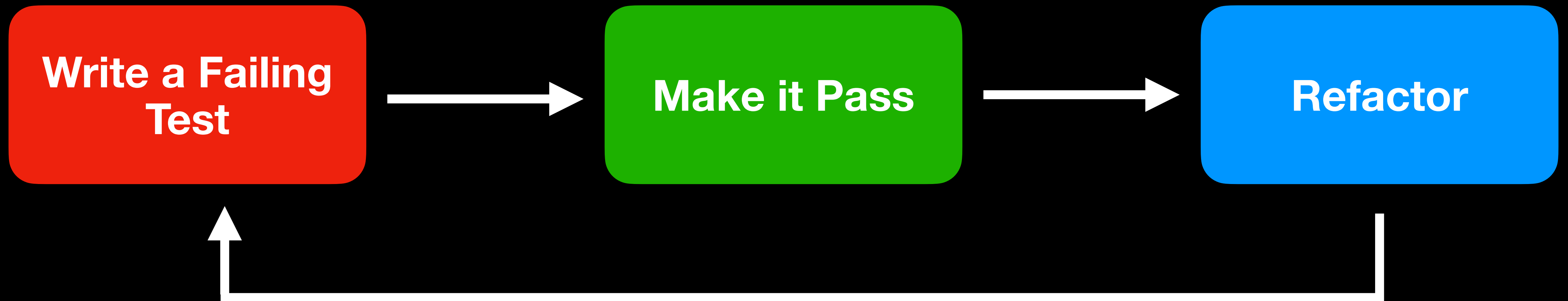
Test-Driven Development

Chapter 1: What is TDD?

조장희

What is TDD?

TDD 주기



What is TDD?

왜 TDD를 사용해야 하는가?

- 소프트웨어가 현재 그리고 미래에도 잘 작동 할 수 있다고 보장 할 수 있습니다.
- 프로덕션 코드를 작성하고 테스트를 통과 시키면 프로덕션 코드가 모든 요구 사항을 충족 하는지 확인이 가능합니다.
- 테스트 코드는 프로덕션의 문서 역할을 대신 할 수 있으며, 이를 통해 작동 방식을 설명 할 수 있습니다.
- 프로덕션 코드와 테스트 코드를 병렬로 반복하여 작성하면서 TDD를 하면 프로덕션 코드는 완전한 테스트를 위해 약간의 리팩토링이 필요하게 되고, 이 리팩토링을 하면 프로덕션 코드와 테스트 코드가 모두 업데이트 되고, 이를 통해 모두 최신 상태를 유지 할 수 있습니다.

What is TDD?

무엇을 테스트 해야 할까?

좋은 테스트란 무엇 일까?

What is TDD?

무엇을 테스트 해야 할까?

범위가 엄청나게 넓은 테스트는 좋은 테스트 일까?

모든 코드를 다 테스트 하는게 좋은 테스트 일까?

도대체 어디서부터 어디까지 테스트 하는게 좋은 걸까?

What is TDD?

무엇을 테스트 해야 할까?

- 대부분의 클래스의 메소드
- 커스텀 된 getter, setter
- 자신이 작성하는 대부분의 코드

자동화 된 방식으로 포착 할 수 없는 코드에 대해 테스트를 하자!

What is TDD?

무엇을 테스트 하지 않아야 할까?

- Generated Code
- 컴파일러에 포착 할 수 있는 error or warning
- 자사 또는 타사 프레임 워크와 같은 dependency(종속성) 코드

What is TDD?

TDD는 오래 걸려!

- TDD는 코드를 더 작성해야 하기 때문에 느리다.
- 익숙하지 않은 방법이기 때문에 적응하는 비용이 더 사용된다.

What is TDD?

정말 TDD는 느리고 비싼 비용이 사용되는 걸까?

- 테스트를 전혀 작성하지 않으면 시간 대비 더 많은 코드를 작성 할 수 있고 처음에는 TDD에 익숙하지 않아 개발 비용이 더 필요합니다.
- 테스트를 하지 않은 코드는 문제가 발생 했을 때 근본적인 원인을 추적하는데 많은 시간 적 비용이 사용 됩니다.
- 시간이 지남에 따라 기능추가, 기존 코드 수정, 버그 수정에 대한 비용이 추가로 발생 하게 되는데 이 때 추가 또는 수정되는 코드들이 영향을 주는 부분을 확인하고 버그를 줄이 는데 많은 비용이 사용 됩니다.
- 버그가 발생 되어 고객이 받는 부정적인 영향과 이에 따른 리뷰, 신뢰 손실을 본다면 장기 적으로는 버그를 줄이고 유지 보수가 더 쉬운 코드를 작성하는 TDD를 하는 것이 테스트 를 하지 않는 것보다 느리고 더 비싼 비용이 사용 된다고 볼 수 없습니다.

What is TDD?

KeyPoint

- TDD는 좋은 테스트를 작성하는 일관된 방법입니다.
- 좋은 테스트는 실패가 되어야 하고, 반복적이며 실행이 빠르고 유지 관리가 가능해야 합니다.
- 관리 책임이 있는 코드에는 테스트를 작성해야 합니다. 자동으로 생성된 generated or code within dependencies를 테스트 하지 않아야 합니다.
- 실제 개발 비용에는 초기 코딩 시간, 사용 중 새로운 기능 추가, 기존 코드 수정, 버그 수정 등이 포함 되는데, TDD는 유지 보수 비용과 버그 수를 줄여서 가장 효율적인 비용으로 접근하는 방식 입니다.
- TDD는 몇개월 이상 지속되거나 여러번 릴리즈 되는 장기 프로젝트에 유용합니다.