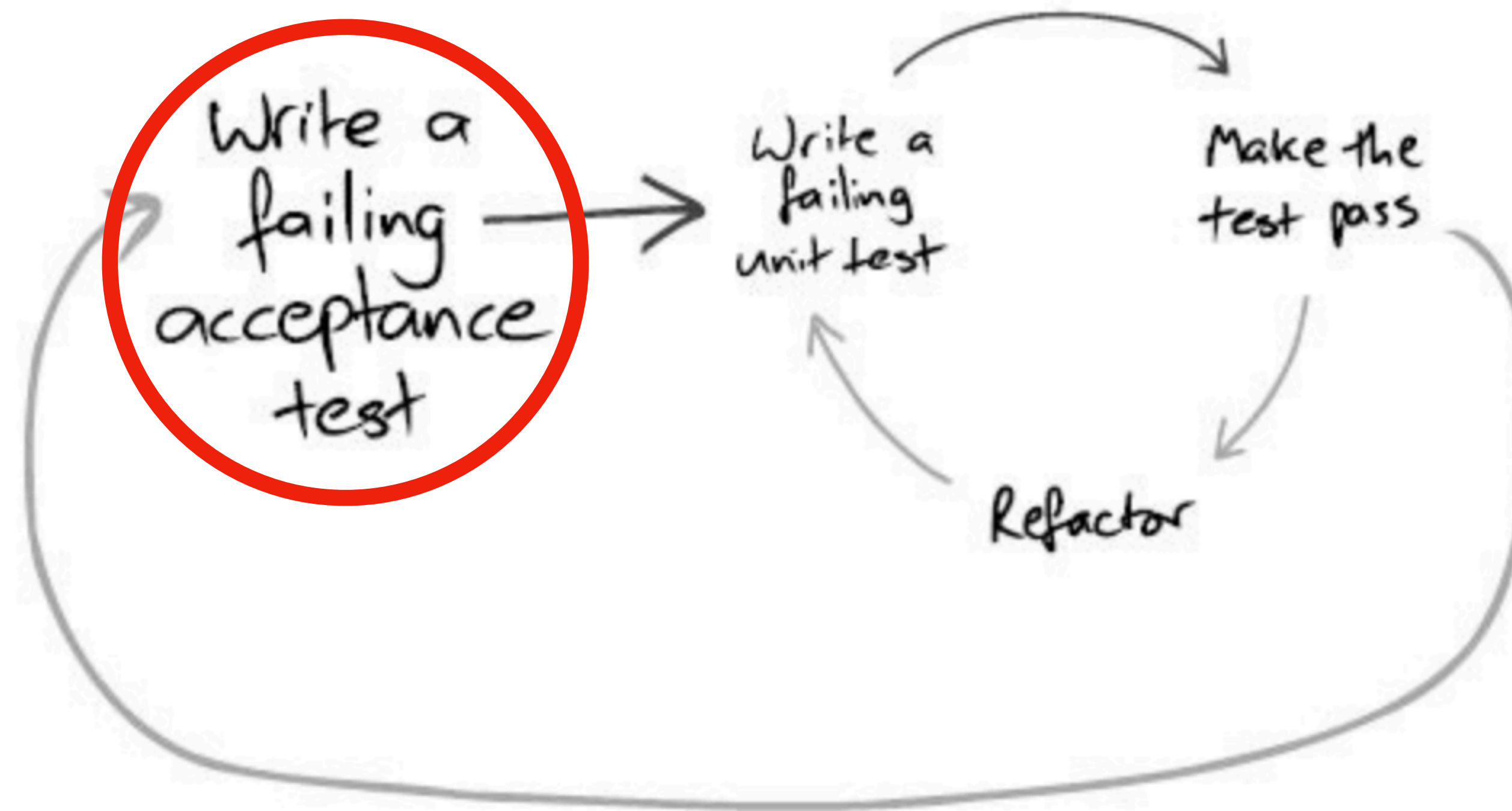


인수 테스트

인수 테스트란?

인수 테스트?



Acceptance Test in extreme programming

- 사용자 스토리를 검증하는 기능 테스트
- 사용자 스토리로 테스트할 시나리오를 지정

Acceptance testing in extreme programming [\[edit \]](#)

Acceptance testing is a term used in [agile software development](#) methodologies, particularly [extreme programming](#), referring to the [functional testing](#) of a [user story](#) by the software development team during the implementation phase.^[13]

The customer specifies [scenarios to test](#) when a user story has been correctly implemented. A story can have one or many acceptance tests, whatever it takes to ensure the functionality works. Acceptance tests are black-box system tests. Each acceptance test represents some expected result from the system.

Acceptance Test in extreme programming

- 사용자 스토리를 검증하는 기능 테스트
- 사용자 스토리로 테스트할 시나리오를 지정

???

Acceptance testing in extreme programming [\[edit \]](#)

Acceptance testing is a term used in agile software development methodologies, particularly extreme programming, referring to the functional testing of a user story by the software development team during the implementation phase.^[13]

The customer specifies scenarios to test when a user story has been correctly implemented. A story can have one or many acceptance tests, whatever it takes to ensure the functionality works. Acceptance tests are black-box system tests. Each acceptance test represents some expected result from the system.

테스트 종류

- 단위 테스트
- 통합 테스트
- E2E 테스트

인수 테스트

- 사용자 스토리를 검증하는 기능 테스트

Acceptance Test

- 명세나 계약의 요구 사항이 충족되는지 확인하기 위해 수행되는 테스트

Acceptance testing

From Wikipedia, the free encyclopedia

In [engineering](#) and its various [subdisciplines](#), **acceptance testing** is a test conducted to determine if the requirements of a [specification](#) or [contract](#) are met. It may involve [chemical tests](#), [physical tests](#), or [performance tests](#).

Acceptance Test

- 소프트웨어 이외 다른 분야에서도 사용되는 용어
- 보통 마지막 단계에서 수행하는 테스트를 의미

국어사전 Beta 인수검사 고급 검색

전체 | 단어 | 속담·관용구 | 뜻풀이 | 예문 | 맞춤법·표기법 T T T ㅁ

인수^검사 引受檢査 + 단어장 저장

1. 군사 현장에 새로 설치된 장비를 실제로 운영하기에 앞서 기능과 성능의 적합성 여부를 살펴보는 일.

머신 비전이란, 산업 현장에서 사람 눈으로 하던 인수 검사를 컴퓨터와 영상 처리 기술로 기판의 불순물이나 스크래치 등 결함을 검사하는 기술을 뜻한다.

출처 <<서울경제 2012년 5월>>

Acceptance testing

From Wikipedia, the free encyclopedia

In **engineering** and its various **subdisciplines**, **acceptance testing** is a test conducted to determine if the requirements of a **specification** or **contract** are met. It may involve **chemical tests**, **physical tests**, or **performance tests**.

Acceptance? 인수?

- Acceptance는 수락, 받아들임이라는 뜻
- Acceptance를 인수로 해석한 이유는 인수를 수락한다는 의미로 해석

Acceptance Test (쉽게 바꾸면)

- 작업을 종료 시켜도 되는지 검증하는 테스트
- 이 테스트가 성공하면 작업 끝!

테스트 주도 개발로 배우는 객체 지향 설계와 실천

우리는 기능을 구현할 때, 만들고자 하는 기능을 수행하는 인수 테스트를 작성하는 것으로 시작한다. 인수 테스트가 실패하는 동안은 시스템이 아직 그 기능을 구현하지 않고 있다는 것을 보여준다; **인수 테스트가 통과되면, 기능 구현은 끝이다.**



콘솔 애플리케이션 개발을 위한 시나리오 기반 인수 테스트 만들기

로또 요구사항

- 콘솔 기반의 로또 애플리케이션을 만든다.
- 금액을 입력하면 금액에 맞는 갯수의 로또를 구입한다.
- 지난주 당첨 번호를 입력하면 당첨 결과와 수익률이 계산된다.

정상적인 시나리오

- 10000원을 입력한다
- 10장의 로또가 구매된다
- 구입한 10장의 로또 번호가 출력된다
- 지난주 당첨 번호로 1, 2, 3, 4, 5, 6을 입력한다
- 당첨 통계가 출력된다
- 수익률이 출력된다

비정상적인 시나리오 - 적은 금액

- 100원을 입력한다
- 로또를 구입할 수 없다.

비정상적인 시나리오 - 유효하지 않은 당첨 번호

- 10000원을 입력한다
- 10장의 로또가 구매된다
- 구입한 10장의 로또 번호가 출력된다
- 지난주 당첨 번호로 1, 1, 1, 1, 1, 1을 입력한다
- 당첨 통계와 수익률을 출력할 수 없다

로또 테스트 코드 작성 - 비정상적인 시나리오 (적은 금액)

```
@DisplayName("로또 한장의 가격보다 적은 금액 입력")
@Test
void lessPriceThanLotto() {
    LottoApplication lottoApplication = new LottoApplication();

    assertThrows(RuntimeException.class, () -> {
        lottoApplication.insertMoney(100);
    });
}
```

로또 테스트 코드 작성 - 비정상적인 시나리오 (유효하지 않은 번호)

```
@DisplayName("유효하지 않은 로또 당첨 번호")
@Test
void invalidWinningLottoNumber() {
    LottoApplication lottoApplication = new LottoApplication();

    Message message = lottoApplication.insertMoney(10000);
    assertThat(message.getText()).isEqualTo("10장의 로또가 구매되었습니다.");

    assertThrows(RuntimeException.class, () -> {
        lottoApplication.insertWinningLottoNumber(1, 1, 1, 1, 1, 1);
    });
}
```

로또 테스트 코드 작성 - 정상적인 시나리오

```
@DisplayName("정상 동작 시나리오")
@Test
void common() {
    LottoApplication lottoApplication = new LottoApplication();

    Message buyMessage = lottoApplication.insertMoney(10000);
    assertThat(buyMessage.getText()).isEqualTo("10장의 로또가 구매되었습니다.");

    Message winningMessage = lottoApplication.insertWinningLottoNumber(1, 2, 3, 4,
    assertThat(winningMessage.getText()).contains("당첨 통계");
    assertThat(winningMessage.getText()).contains("당첨 수익률");
}
```

API 개발을 위한

시나리오 기반 인수 테스트 만들기

요구사항: 지하철역 관리 기능 구현하기

- 지하철 역 생성
- 지하철 역 목록 조회
- 지하철 역 삭제

시나리오: 지하철역 관리 기능 구현하기

- 지하철 역 생성
 - 지하철 역 생성 요청을 한다.
 - 지하철 역이 생성을 되었다.
- 지하철 역 목록 조회
 - 지하철 역이 생성되어 있다.
 - 지하철 역 목록 조회 요청을 한다.
 - 지하철 역 목록이 응답되었다.

...

API 레벨의 테스트 작성

```

@DisplayName("지하철역을 생성한다.")
@Test
void createStation() {
    // given
    Map<String, String> params = new HashMap<>();
    params.put("name", "강남역");

    // when
    ExtractableResponse<Response> response = RestAssured.given().log().all()
        .body(params)
        .contentType(MediaType.APPLICATION_JSON_VALUE)
        .when()
        .post("/stations")
        .then().log().all()
        .extract();

    ...
}
```


테스트 응답 결과를 검증

```
import static org.assertj.core.api.Assertions.assertThat;

...

@DisplayName("지하철역을 생성한다.")
@Test
void createStation() {

    ...

    ExtractableResponse<Response> response = RestAssured.given().log().all()
        .body(params)
        .contentType(MediaType.APPLICATION_JSON_VALUE)
        .when()
        .post("/stations")
        .then().log().all()
        .extract();

    // then
    assertThat(response.statusCode()).isEqualTo(HttpStatus.CREATED.value());
    assertThat(response.header("Location")).isNotBlank();
}
```

이번 과정에서의 인수 테스트

인수 테스트는 API 테스트?

인수 테스트는 E2E 테스트?

인수 테스트는 통합 테스트?

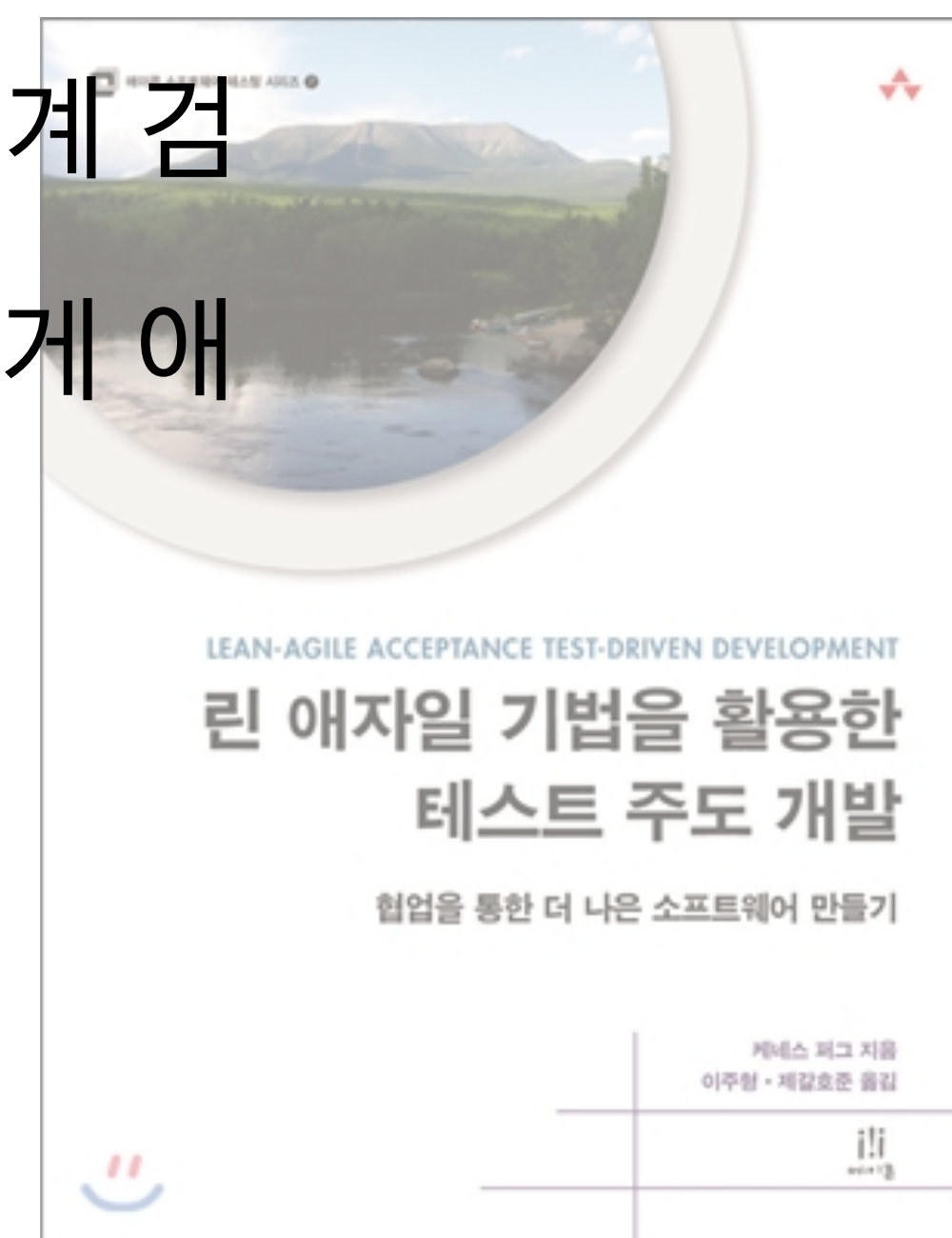
인수 테스트는
테스트의 의도에 따라
구현 방법이 달라진다.

테스트가 검증하는 대상

- 단위 테스트: 구현한 부분, 단위를 검증
- 통합 테스트: 각 단위들이 유기적으로 잘 동작하는지 검증
- 인수 테스트: 요구사항을 만족하는지를 검증

린 애자일 기법을 활용한 (인수) 테스트 주도 개발

- 인수 테스트의 개념은 테스트 의도에 따라 정해지는 것이지 테스트를 어떻게 구현하는지에 따라 정해지는 것이 아니다. 유닛 레벨이나 통합 레벨, 사용자 인터페이스 레벨에서 인수 테스트를 적용할 수 있다. ... 더 나아가, 인수 테스트를 유닛이나 컴포넌트가 의도한 동작을 하는지 확인하는 설계 검증 테스트로 사용할 수 도 있다. 어떤 경우든 인수 테스트는 사용자에게 애플리케이션이 인도될 수 있는 지를 확인한다.



이번 과정에서의 인수 테스트

- 백엔드 개발자가 단독적으로 적용해서 실천해볼 수 있는 범위
- 고객은 프론트엔드 개발자 혹은 API 활용하는 사람 대상
- API 접점에서 검증하는 **E2E 테스트**
- API의 Request와 Response 정보 이외 내부 정보는 최대한 가리는 **블랙 박스 형식의 테스트**