

# 7과목 (애플리케이션 테스트 관리)

## Chapter 01 애플리케이션 테스트 케이스 설계

### 1. 애플리케이션 테스트

↘ 애플리케이션에 탑재되어 있는 결함을 찾아내는 일련의 행위 또는 절차

### 2. 애플리케이션 테스트 원리

- 완벽한 테스트를 불가능 : 결함을 줄일 수 있으나, 결함이 없다고 증명할 수 없음
- **파레토 법칙(Pareto Principle)** : 20%에 해당하는 코드에서 전체 결함의 80%가 발견된다는 법칙
- **살충제 패러독스(Pesticide Paradox)** : 동일한 테스트를 반복하면 더 이상 결함이 발견되지 않는 현상
- 정황 의존성 : 소프트웨어 성격에 맞게 테스트 실시
- **오류-부재의 귀변** : 요구사항을 충족시켜주지 못한다면, 결함이 없다고 해도 품질이 높다고 볼 수 없음

### 3. 프로그램 실행 여부에 따른 분류

- **정적 테스트** : 테스트 대상을 실행하지 않고 구조를 분석하여 논리성을 검증하는 테스트
  - 리뷰 : 여러 전문가들이 모여 코드를 포함하여 소프트웨어 개발 및 산출물을 검토하고 테스트 하여 결함을 검출하는 방법
    - **동료 검토** : 2~3명이 진행하는 리뷰형태
    - **워크 스루** : 검토 자료를 회의전에 배포해서 사전검토후 짧은 시간 동안 진행하는 형태
    - **인스펙션** : 저작자 외의 다른 전문가 또는 팀이 검사
  - 정적 분석 : 자동화된 도구를 활용하여 정적 테스트를 수행하여 결함을 발견한다.
- **동적 테스트** : 소프트웨어를 실행하는 방식으로 테스트를 수행하여 결함을 검출하는 테스트
  - **화이트박스 테스트** (구조기반 테스트): 구현된 소스코드와 내부 구조를 바탕으로 테스트 케이스를 결정하는 방법 (조루데기)
    - 제어 구조 검사
      - **조건 검사**: 프로그램 모듈 내에 있는 논리적 조건을 테스트
      - **루프 검사**: 프로그램의 반복(Loop) 구조에 초점을 맞춰 실시하는 테스트
      - **데이터 흐름 검사**: 프로그램에서 변수의 정의와 변수 사용의 위치에 초점을 맞춰 실시하는 테스트
    - **기초 경로 검사**: 테스트 케이스 설계자가 절차적 설계의 논리적 복잡성을 측정할 수 있게 해주는 테스트 기법
  - **블랙박스 테스트** (명세기반 테스트) : 사용자의 요구사항 명세를 보면서 수행하는 테스트 (동경원오비)
    - **\*\*\*동등 분할 테스트\*\*\*** : 입력 데이터의 영역을 유사한 도메인별로 유효값/무효값을 그룹핑하여 대푯값 테스트 케이스를 도출하여 테스트 하는 기법

ex ) 90~100 테스트 중간 값 95 대표값

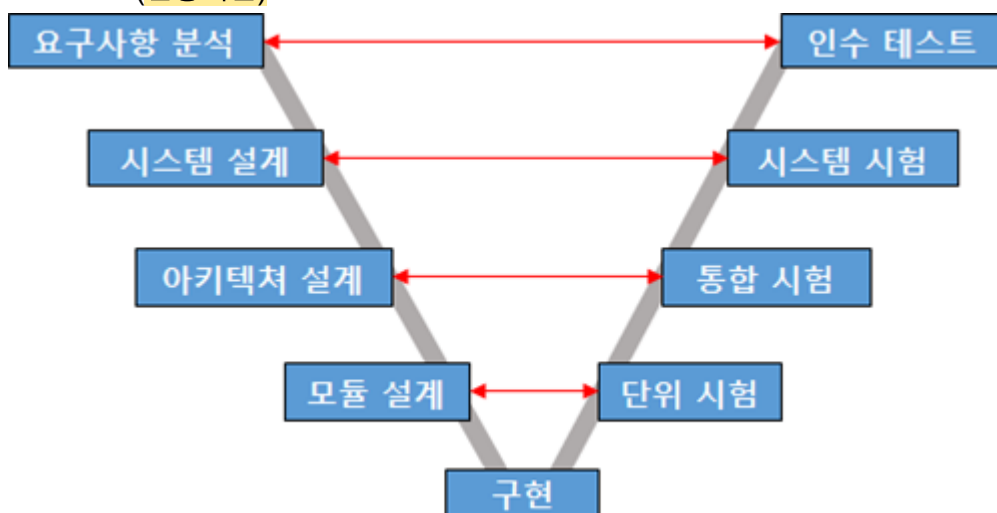
- **경계값 분석 테스트** : 입력 조건의 경계값을 테스트 케이스로 선정하여 검사하는 기법  
ex ) 90~100 테스트 89또는 101점 테스트
- **원인-결과 그래프 테스트** : 입력값을 원인으로, 효과를 출력값으로 선정하는 테스트 기법
- **\*\*\*오류예측 검사\*\*\*** : 과거의 경험이나 확인자의 감각으로 테스트하는 기법
- **비교 테스트** : 여러 버전의 프로그램에 동일한 테스트 자료를 제공하여 동일한 결과가 출력되는지 테스트

#### 4. 테스트 커버리지

↳ 소프트웨어의 소스코드가 얼마만큼 테스트 되었는가를 나타내는 지표

- 구문(문장) 커버리지 : 프로그램 내의 **모든 명령문을 적어도 한 번 수행**하는 커버리지
- 결정(분기) 커버리지 : 결정 포인트 내의 **전체 조건식**이 적어도 한번은 참과 거짓의 결과가 되도록 수행하는 커버리지
- 조건 커버리지 : 결정 포인트 내의 각 **개별 조건식**이 적어도 한번은 참과 거짓의 결과가 되도록 수행하는 커버리지
- 조건/결정 커버리지 : **전체 조건식 & 개별 조건식 모두** 참 한번, 거짓 한 번 결과가 되도록 수행하는 커버리지

#### 5. V-Model (단통시인)



- **단위 시험** : 분리하여 시험 가능한 소프트웨어 모듈의 기능을 확인하는 활동
- **통합 시험** : 모듈을 결합하여 하나의 시스템으로 구성시 테스트 수행
- **시스템 시험** : 통합 모듈에 대한 시스템적 테스트, 신뢰성, 견고성, 성능, 안전성등의 비기능적 요구사항도 테스트
- **인수 시험** : 사용자의 만족여부를 테스트
  - 알파 테스트 : 개발자 장소에서 개발자와 함께
  - 베타 테스트 : 실제 사용자에게 소프트웨어 사용하게 함

#### 6. 테스트 시각에 따른 분류

- 검증 : 소프트웨어 개발 과정을 테스트, 개발자 시각
- 확인 : 소프트웨어 결과를 테스트, 사용자 시각

#### 7. 테스트 목적에 따른 분류 (회안성 구회병)

- 회복 테스트 : 고의로 실패 유도, 정상적 복귀 테스트
- 안전 테스트 : 보안적인 결함 미리 점검하는 테스트
- 성능 테스트 : 응답시간, 처리 속도 등 측정 테스트 (부스스내)
  - 부하 테스트 : 부가를 증가시켜 임계점을 찾는 테스트
  - 스트레스 테스트 : 임계점 이상의 부하를 가하여 비정상적인 상황에서의 테스트
  - 스파이크 테스트 : 짧은 시간에 사용자가 몰릴 때 시스템의 반응 측정 테스트
  - 내구성 테스트 : 오랜 시간 동안 시스템에 높은 부하를 가해 테스트
- 구조 테스트 : 내부 논리 경로, 소스코드 복잡도 평가 테스트
- 회귀 테스트 : 이미 테스트가 완료된 소프트웨어에 변경 또는 수정된 코드에 새로운 결함이 없음을 확인하는 테스트
- 병행 테스트 : 변경된 시스템과 기존 시스템 동일한 데이터 입력 후 결과 비교 테스트

## 8. 통합 테스트

- 하향식 통합방법
  - ↳ 메인제어 모듈(프로그램)부터 아래방향으로 제어의 경로를 따라 하향식으로 통합하면서 테스트 진행, ==스텝==사용
- 상향식 통합 방법
  - ↳ 최하위 레벨의 모듈 또는 컴포넌트부터 위측방향으로 제어의 경로를 따라 이동하면서 테스트 수행, 드라이버 사용

## 9. 테스트 오라클 종류

- 참(True) 오라클 : 모든 입력값에 대하여 기대하는 결과를 제공하는 오라클
- 샘플링(Sampling) 오라클 : 특정한 몇 개의 입력값에 대해서만 기대하는 결과를 제공해주는 오라클
- 휴리스틱(Heuristic) 오라클 : 특정 입력값에 대해 올바른 결과를 제공하고, 나머지 값들에 대해서는 추정(Heuristic)으로 처리하는 오라클
- 일관성 검사(Consistent) 오라클 : 애플리케이션 변경이 있을 때, 수행 전과후의 결과값이 동일한지 확인하는 오라클

## 10. 애플리케이션 성능 측정 지표

- 처리량(Throughput) : 애플리케이션이 주어진 시간에 처리할 수 있는 트랜잭션의 수
- 응답 시간(Response Time) : 사용자 입력이 끝난 후, 애플리케이션의 응답 출력이 개시될 때까지의 시간
- 경과 시간(Turnaround Time) : 애플리케이션에 사용자가 요구를 입력한 시점부터 트랜잭션을 처리후 그 결과의 출력이 완료될 때까지 걸리는 시간
- 자원 사용률(Resource Usage) : 애플리케이션이 트랜잭션을 처리하는 동안 사용하는 CPU, 메모리, 네트워크 사용량