

# Software Testing

## - 소프트웨어 테스트

- Seven Testing Principles

- 완벽한 테스트는 불가능하다
- 결함 집중(Defect Clustering): 대부분의 결함은 적은 수의 모듈(기능)에서 발견된다.
- 소프트웨어에 결함이 없다고 증명할 수 없다.
- 살충제 페러독스: 동일한 테스트를 반복적으로 수행하면 버그를 찾기 힘들
- 오류-부재의 귀변 : 오류가 거의 없어도 소비자가 요구한 것과 다르면 의미가 없다
- 빠른 테스트 : 테스트는 소프트웨어 개발 주기에서 가능한 빨리 시작되어야한다.
  - 테스트는 상황에 의존적이다
  - 테스트는 상황에 따라 다르게 진행: 의료기기는 테스트가 훨씬 잘 되어야한다

- SDLC vs STLC

- Waterfall 모델을 사용하면 다시 돌아가는 과정에서 많은 비용이 발생하게된다.
- Testing을 빠르게 진행할 수록 비용이 적게 든다
- SDLC: 요구명세 -> 분석 -> 상위 설계 -> 하위 설계 -> 코딩
- STLC: Unit Test(단위) -> Integration Test(통합) -> System Test(시스템) -> Acceptance Test(인수)
- 점진적 개발(incremental development)
- The Agile Scrum

- Unit Testing

- 개발자가 시행
- ex) 로그인 모듈을 테스트할 경우
  - 유효한 아이디 & 비밀번호 입력
  - 유효하지 않은 아이디 & 비밀번호 입력
  - 아이디를 넣지 않고 로그인 버튼 클릭
  - SQL Injection 등등

- Integration Testing

- 통합 테스트는 테스트팀이 개발자와 함께 수행
- BigBang 방식
  - 모든 모듈이 개발될 때까지 대기
  - 어디서 오류가 났는지 찾기 어렵다
- Incremental Testing
  - 사용이 가능한 모듈부터 테스트

- System Testing

- 시스템 테스트는 전체 시스템의 동작과 관련이 있습니다
- 사용자가 사용하는 방식으로 End to End 시나리오로 테스트합니다
- 기능 외에 비기능 요구사항도 확인해야 합니다 (Nonfunctional requirement)

- Acceptance Testing
  - 티몬에서는 사용하지 않는다
  - 고객이 개발사에 프로그램을 요청했을때 시행하는 과정
  - 요청사의 요구사항에 맞게 개발이 되었는지를 확인하는 절차
    - 알파 테스트 - 비공개 내부 테스트, 소수의 직원, 관계자
    - 베타 테스트 - 외부의 소수 사용자
- Non-Functional Testing
  - LoadRunner라는 성능테스트 툴이 있다

## - 테스트 기법

- 동적 테스트
  - 블랙박스 테스트
    - 프로그램 외부 명세를 보면서 테스트(기능테스트)
    - 동등 분할, 경계값 분석 등등
      - 동등분할: 각 부분의 값을 각각 테스트
      - 경계값분석: 경계부분에 오류가 많다는 가정하에 테스트
      - 결정테이블: 입력의 조합으로 다른 결과를 만드는 테이블을 작성하는 기법
      - 구문 테스트: 테스트케이스 Suite에 의하여 실행된 구문이 몇 %인지 측정
      - 분기 테스트: 테스트 케이스 Suite에 의하여 실행된 조건문 분기가 몇 %인지 측정
      - 경험 기반 테스트: 탐색적 테스트를 주로한다. 명세가 거의 없고 시간이 부족한 경우 적합하다.
  - 화이트박스 테스트
    - 프로그램 내부 로직을 보면서 테스트(구조테스트)
- 정적 테스트
  - 리뷰
    - 리뷰는 간단히 말해서 사람들이 소프트웨어 작업 전에 제품을 분석하고 품질향상을 위해서 변경을 권고하는 회의 입니다.
    - 설계문서, 요구사항 문서, 개발 코드, 테스트 계획 등 중간 산출물은 리뷰의 대상이 될 수 있다.
    - 리뷰의 필요성
      - 테스트 보다는 Review에서 결함을 발견하는 것이 더 효율적
      - 단위 테스트 후의 결함제거에는 많은 비용이 필요
      - 빠른 결함 제거시 시간과 비용 절감 효과가 있다
    - 리뷰의 성공요소
      - 리뷰가 명확하게 미리 정의된 목적이 있어야함
      - 결함 발견은 언제나 환영하는 분위기여야함
      - People issues와 심리적인 측면이 고려되어야 함
      - 리뷰 기법이 적절히 적용되어야 함
- 코드를 개발하면 단위 테스트를 잘하고 만족되면 Commit을 해라!!!