**소프트웨어 테스팅의 필요성**

* **소프트웨어 시스템의 중요성**: 소프트웨어 시스템은 생활의 많은 부분에서 사용되고 있으며, 그 비중은 계속해서 증가하고 있다.
* **테스팅의 필요성**: 소프트웨어가 올바르게 동작하지 않는 경우 다양한 문제가 발생할 수 있으며, 이로 인해 **금전적 손실**, **시간 낭비**, **비즈니스 이미지 손상** 등의 피해가 발생할 수 있다.
* **테스팅의 목적**: 테스팅은 이러한 문제를 최소화하기 위해 반드시 필요하다.

**결함의 원인과 정의**

* **결함 발생 원인**: 결함은 사람이 오류를 범하기 쉽기 때문에 발생하며, 다음과 같은 여러 요인으로 인해 발생한다.
  1. **시간적인 압박**
  2. **복잡한 코드**
  3. **기반 환경의 복잡성**
  4. **기술이나 시스템의 변경**
  5. **시스템 간의 연동**
* **장애의 정의**: 장애는 결함에 의해 발생할 수 있으며, 환경적인 조건에 의해서도 발생할 수 있다.
* **용어 정의**:
  1. **오류(Error)**: 결함이 되는 요소.
  2. **결함(Defects, Bug)**: 에러가 소프트웨어상에 나타나는 것.
  3. **장애(Failure)**: 시스템이 의도된 대로 동작하지 않거나 동작하지 말아야 함에도 동작함.

**테스팅의 역할과 중요성**

* **결함 발견 및 수정**: 테스팅을 통해 발견하지 못했던 결함들을 릴리즈 전에 발견하고 수정함으로써 운영 환경 내에서 발생하는 결함들의 리스크를 줄일 수 있다.
* **품질 향상**: 테스팅은 소프트웨어 시스템의 품질 향상에도 기여한다.
* **테스팅의 시작 시점**: 개발 초기의 요구사항 분석 단계부터 리뷰와 정적 분석을 통해 테스팅이 시작될 수 있다.
* **테스트 레벨**: 각 개발 단계에 대응하는 테스트 레벨에 따라 테스팅이 이루어진다.

**테스팅의 원리**

* **원리 1**: 테스팅은 결함이 존재함을 밝히는 활동이다. 결함이 전혀 발견되지 않은 경우에도 해당 소프트웨어에 결함이 없다고 증명할 수는 없다.
* **원리 2**: 완벽한 테스팅(Exhaustive testing)은 불가능하다. 이는 다음과 같은 이유로 설명된다.
  1. 한 프로그램 내의 내부 조건이 무수히 많을 수 있음 (무한 경로)
  2. 입력이 가질 수 있는 모든 값의 조합이 무수히 많음 (무한 입력값)
  3. GUI 이벤트 발생 순서에 대한 조합도 무수히 많음 (무한 타이밍)
* **원리 3**: 테스팅은 개발 초기에 시작해야 하며, 설정한 테스트 목표에 집중해야 한다.
* **원리 4**: 결함 집중(Defect clustering) - 대다수의 결함들은 소수의 특정 모듈에 집중되어 발생하는 경향이 있다.

**테스트 프로세스 개요**

* **테스트 계획 수립**: 테스트 목표와 임무를 달성하기 위해 면밀히 확인하고 필요한 활동을 정의하는 것이다.
* **주요 작업 내용**:
  1. **테스트 범위**: 다른 시스템과의 인터페이스 정고, 관련 품질 특성, 호환성 테스팅 범위 등.
  2. **리스크 기반 테스트 전략**: 각 테스트 레벨에 대한 테스트 대상 제품이 충족해야 할 품질 수준 및 특성을 고려한 테스트 전략 수립.
  3. **테스팅 목적**: 품질 요구 수준, 테스트 레벨 별 목적(결함 발견, 요구사항 충족 등).
* **테스트 제어**: 계획 대비 실제 진행 상황을 지속적으로 비교하는 활동이다.

**테스트 분석과 설계**

* **테스트 분석과 설계의 정의**: 일반적이고 추상적인 테스팅 목적을 실제적이고 구체적인 테스트 상황과 테스트 케이스로 변환하는 활동이다.
* **주요 작업**:
  1. **테스트 베이시스 리뷰**: 요구사항 명세서, 아키텍처, 개발 설계 문서, 인터페이스 등을 리뷰한다.
  2. **테스트 상황 식별**: 테스트 대상 아이템 또는 제품, 명세, 동작과 구조의 분석을 통해 테스트 상황을 식별하고 우선순위를 선정한다.
  3. **테스트 데이터 식별**: 테스트 상황과 테스트 케이스에 필요한 테스트 데이터를 식별한다.

**테스트 실행과 완료 조건**

* **테스트 실행의 정의**: 가장 효율적이고 효과적으로 테스트를 실행하기 위해 테스트 케이스를 조합하고 테스트 실행에 필요한 다른 정보를 포함하는 테스트 프로시저 또는 테스트 스크립트를 명세화하는 활동이다.
* **주요 작업**:
  1. **테스트 케이스 개발**: 테스트 케이스의 개발 및 구현과 우선순위 선정.
  2. **자동화 테스트 스크립트 작성**: 테스트 하네스 준비 및 테스트 케이스 묶음(Test suites) 생성.
  3. **결과 확인**: 예상 결과와 실제 결과 간의 차이에서 오는 불일치를 incidents 또는 결함으로 보고한다.
* **완료 조건 평가**: 초기에 정의된 테스트 목표에 비해 어느 정도 실제 테스트가 수행되었는지를 평가하는 활동이다.

**테스트 마감 활동**

* **테스트 마감의 정의**: 완료된 테스트 활동에서 데이터를 수집하여, 테스트에서 발견된 사실 및 수치적 데이터와 함께 테스팅 경험과 테스트 웨어를 종합하고 축적하는 활동이다.
* **주요 작업**:
  1. **테스트 결과 마감**: 테스트 결과를 정리하고 보관하는 것.
  2. **테스트 프로세스 심사**: 얼마나 체계적으로 수행되었는지를 평가하고 향후 테스팅을 개선하기 위한 모범 사례 모델링.
  3. **테스트 웨어 이관**: 테스트 웨어를 유지보수 조직에 이관한다.