**1. 테스트는 결함이 존재함을 밝히는 활동이다**

소프트웨어에 대해 테스트 완료 및 발견된 이슈를 모두 해결하여도 결함이 없다는 것을 증명할 수 있는 것은 아니다. 또한 이슈가 발견되지 않았다고 해서 결함이 없다는 것이 증명되지는 않는다. 테스트는 프로그램의 결함이 없음을 보장하는 활동이 아니라, **결함이 존재함을 밝히기 위한 활동**이다.

**2. 완벽한 테스팅은 불가능하다**

매우 단순한 소프트웨어가 아닌 이상 **내부 조건, 입력값, 타이밍에 대한 모든 조합**을 확인할 수 없다. 따라서 테스트 대상의 리스크 분석 후에 **가장 중요한 부분을 중점으로 테스팅 리소스를 투입**하여야 한다.

**3. 테스팅은 개발 초기 단계에서부터 시작해야 한다**

요구 사항 분석 및 설계 단계에서부터 테스트를 진행하는 경우 **문서상의 결함**을 확인할 수 있다. 이러한 결함은 **코딩 작업 이후에 발견되는 결함**에 비해 훨씬 간단하게 해결할 수 있다. 또한 조기에 테스트 설계를 마친 경우 코딩이 완료되자마자 테스트 케이스를 레벨별로 실행할 수 있어 **전체 테스트 기간을 단축**할 수 있다.

**4. 결함 집중**

대다수의 결함은 **소수의 특정 모듈에 집중**되는 경향이 있고, 이러한 결함은 장애를 발생시킬 가능성이 높다.

* 자체적으로 복잡한 구조를 가지고 있는 모듈
* 소프트웨어의 다른 부분과 복잡한 상호 작용을 하는 모듈
* 개발 난이도가 높거나 최신 기술을 사용한 모듈
* 기존의 것을 사용하지 않고 새롭게 개발한 모듈
* 크기가 큰 모듈
* 경험이 미흡한 개발팀에서 개발한 모듈

**5. 살충제 패러독스**

**동일한 테스트 케이스를 반복적으로 수행하는 경우 더 이상 새로운 결함을 찾아낼 수 없다.** 이를 극복하려면 새로운 기법, 다른 시각에서 **테스트 케이스를 정기적으로 변경하고 추가해주어야 한다.**

**6. 테스팅은 정황에 따라 이루어져야 한다**

소프트웨어의 **종류나 목표 등에 따라 해당 소프트웨어에 맞는 테스트 방식이 적용**되어야 한다. 개발 프로젝트인지 운영중인 시스템의 유지보수인지 여부, 사용 가능한 예산, 출시 일정, 리스크, 조직 문화, 사용자의 기대, 테스팅에 필요한 기반 환경의 가용성, 프로젝트의 중요도 등이 고려되어야 한다.

**7. 오류 부재의 귀변**

거의 모든 결함을 확인 후 제거하였다고 해도 **사용자의 요구 또는 비즈니스 목적을 충족시키지 못하는 경우 품질이 높다고 할 수 없다.**

출처 https://kairoka-sqa.tistory.com/1