

# 결함의 분류

이창공

2023년 11월 9일

## Abstract

결함의 분류는 중요한 문제이다. 알맞은 수정 방법을 찾는 것 또한 중요한 문제이다. 이러한 수정 방법은 결함 분류의 좋은 기준이 될 수 있다.

결함의 분류는 중요한 문제이다. 개발자는 결함 양산기와 같다. 결함이 존재하는 코드를 작성해본 경험이 없는 개발자는 없으며 결함 발견 및 수정의 자동화 기술의 수요가 꾸준하다는 사실이 이를 뒷받침 한다. 직책의 의미와 일맥상통하듯 개발자의 의의는 소프트웨어의 개발에 있다. 따라서 유지보수를 위한 코드의 가독성, 유지 보수성, 확장성은 개발자의 의무이자 책임이다. 안타깝게도 결함은 끊임없이 생산되며 개발보다 유지보수에 비용이 많이 사용되는 실정이다. 만약 양산되는 결함의 원인과 수정 방법을 분류하고 이를 자동화할 수 있다면 큰 비용적인 이점이 존재하는 것은 자명하다.

알맞은 수정 방법을 찾는 것 또한 중요한 문제이다. 하나의 결함에 여러 수정 방법이 존재할 수 있지만, 이 중 가장 좋은 수정 방법을 결정하는 요건에는 여러 가지가 있다. 예를 들어, 결함의 수정은 다른 결함을 일으켜서는 안되며 프로그램의 의도를 고려하여야 한다. 예를 들어, 사용자의 입력 때문에 결함이 발생하면 실행을 종료하는 수정을 했을 때 할당된 메모리 반환을 하지 않는다면 메모리 누수가 발생할 수 있다. 또한 프로그램의 의도가 실행 종료와 맞지 않다면, 결함을 우회하는 방법이 사용되어야 할 것이다.

수정 방법은 결함 분류의 좋은 기준이 될 수 있다. 예를 들어, 보안 취약점에서는 가능한 공격의 방법이 분류의 중요한 조건이 될 것이다. 하지만 일반적으로 결함의 원인이 분류의 기준으로 사용하고 있다. 따라서 같이 분류되는 버퍼 오버플로 취약점이라도, 초과한 버퍼의 미세한 양 차이에 따라서 공격 방법이 달라질 수 있으며 공격자가 원하는 코드를 실행시킬 수 있는지가 달라진다. 또한 정수 오버플로 취약점으로 분류된 취약점이 버퍼 오버플로 등의 다른 결함을 일으킬 수도 있다. 다른 관점에서도 마찬가지이다. 제어 흐름, 데이터 흐름, 프로그램의 의미와 개발 방향 등 여러 가지 측면에서 같은 분류에 속하는 결함일지라도 분류 목적에 따라 장단점이 존재할 수 있다. 반면에 수정 방법을 고려하여 결함을 분류하는 것도 가능할 것이다. 과거에 존재했던 결함이 수정된 기록이 있다면 같은 방법으로 수정이 가능한 결함들을 묶어서 유사한 결함으로 분류하는 것이다. 만약 같은 프로그램에서 과거의 수정 방법이 적용 가능한 결함이 발생이 존재하고 분류할 수 있다면, 결함의 발견과 수정의 단계가 단축될 뿐만 아니라 기준이 되는 수정 방법의 품질에 따라 부작용이 적고 프로그램의 의도를 고려한 수정이 가능할 것이다.