



## 테스트 개요

테스트는 처음부터 프로그래밍과 함께였습니다. 여러분도 첫 프로그램을 작성하고 나서 분명 샘플 데이터를 입력하여 코드가 제대로 동작하는지 살펴보았을 것입니다. 하지만 오랫동안 소프트웨어를 테스트하는 프로세스는 크게 개선되지 못했습니다. 대부분 수동으로 이루어졌고 오류가 나기 쉬운 프로세스였고, 하지만 2000년대 초부터 소프트웨어 업계는 덩치가 커진 소프트웨어 시스템의 복잡도에 대응하기 위해 테스트 방식을 극적으로 진화시켰습니다. 그 진화의 중심에는 개발자가 주도하는 테스트와 자동 테스트가 있었습니다.

**자동 테스트** automated test는 버그가 몰래 숨어들어 고객을 놀라게 하는 사태를 막아줍니다. 개발 주기에서 버그를 발견하는 시기가 늦어질수록 고치는 비용이 커집니다. 많은 경우 기하급수적으로 커지죠.<sup>1</sup> 하지만 '버그 잡기'는 테스트를 하는 여러 이유 중 하나일 뿐입니다. 예를 들어 '소프트웨어가 변화할 수 있도록 지원'하는 역할 역시 버그 잡기에 못지않게 중요합니다. 새로운 기능을 추가하거나, 코드가 더 견실해지도록 리팩터링하거나, 대규모 재설계를 진행하는 상황에서 자동 테스트는 실수를 빠르게 잡아주므로 안심하고 소프트웨어를 변경할 수 있게 해줍니다.

반복 주기가 짧은 변화하는 기술, 시장 상황, 고객 취향을 더 빠르게 받아들일 수 있습니다. 테스트 체계가 잘 갖춰져 있다면 변화를 두려워할 이유가 없습니다. 따라서 테스트 체계

1 (출처: 구글) 'Defect Prevention: Reducing Costs and Enhancing Quality' 참고, <https://oreil.ly/27R87>