

설득을 자동화하기

장봉준

아무리 성능이 뛰어난 버그탐지 도구라도 정작 개발자를 설득하지 못한다면 그 가치를 인정받기 어렵다. 개발자의 실수를 방지하고 소프트웨어를 견고하게 하고자 하는 목적은 분명하지만, 실제로는 도구들이 개발자들에게 무시당하거나 신뢰를 잃는 사례가 빈번하다. 자동 버그 탐지가 정말로 빛을 발하려면 개발자들을 설득해야만 한다. 이 글에서는 도구를 평가하는 새로운 방법을 제시함으로써 버그탐지 도구의 가치를 재고하는 방법에 대해 논의하고자 한다.

소프트웨어 버그탐지 도구들은 개발환경 통합에 더욱 집중해야 한다. 개발환경 통합이란 도구가 IDE, 빌드 시스템, 코드리뷰 과정에 자연스럽게 녹아드는 것을 말한다. 개발환경에 통합되어 개발자가 고품질 코드를 작성하도록 지속적으로 설득할 수 있어야만 이런 도구들은 그 가치를 인정받을 것이다. 하지만 도구의 표면적 성능만을 우선하는 작금의 행태는 도구 개발과 현실 세계의 간극을 더욱 심화시키고 있다.

소프트웨어 결함을 자동으로 탐지하는 도구들의 성능은 나날이 발전하고 있다. 더 빨리 더 많이 더 정확히 버그를 잡아내는 퍼저와 정적분석기들이 계속해서 등장하고 있으며, 지향성 퍼저처럼 더 발전된 형태의 도구들도 나오고 있다. 인공지능을 활용하는 기술도 새롭게 탄생하고 있으니 자동 버그탐지 기술이 계속해서 발전하리라는 예측에는 의심이 없다.

그러나 도구의 발전에도 불구하고, 개발 현장에서의 실용성은 여전히 의문이다. 개발자들이 새로운 도구를 적극적으로 활용하고 있지 않기 때문이다. 개발자의 작업 과정과 유기적으로 연결되지 않으면 아무리 강력한 기능이라도 외면받을 수 밖에 없다. 스스로 지상최고라고 주장하는 도구(State-of-the-art)조차도 개발자들이 쉽고 유용하다고 느끼지 못하면 곧 수명을 다하고 아무도 사용하지 않는 도구가 되고 만다.

우리는 도구의 성능을 개발자의 관점에서 평가하는 새로운 접근이 필요하다. 더 빠르고 정확한 도구를 만드는 것이 아니라 개발자들이 더 쉽게 받아들일 수 있도록 돕는 것에 초점을 맞춰야 한다. 지금까지는 정해진 시간안에 더 적은 계산자원으로 버그를 탐지하는 것을 목표로 했으나, CPU와 메모리는 더 이상 비싼 자원이 아니며 이를 최적화하는 것은 큰 의미가 없다.

개발환경에서 가장 비싼 것은 이제 계산자원이 아니라 개발자 인지부하(Cognitive Load)이다. 인지부하란 개발자가 코드를 이해하거나 작성하는 데 필요한 노력의 양이다. 그리고 버그를 이해하는 일은 보통 적지 않은 인지부하를 요구한다. 예를 들어 보자. 개발자는 보통 이미 리뷰가 끝났거나 배포가 완료된 코드는 인지부하가 많이 필요하다고 생각하지 않는다. 이런 코드에서 버그 정보가 발생하면, 개발자는 이를 중요하게 여기지 않을 가능성이 높아진다. 이러한 사례가 반복된다면 개발자는 도구를 유용하다기보다 귀찮다고 느끼게 된다.

알맞은 상황에 적은 인지부하로 개발자를 설득하는 것이 도구의 덕목이 되어야 할 것이다. 구글과 애플의 보고서에서 이를 확인할 수 있다. 구글에서는 코드 리뷰 단계보다 컴파일 과정에서 버그 정보를 보여줄 때 개발자들이 이를 더 중요하게 받아들인다는 사실을 발견했다. 따라서 최대한 컴파일 과정에 정보를 띄움으로써 개발자들이 버그없는 코드를 작성하도록 돕는 방법을 제시했다. 개발자가 높은 인지부하가 필요하다고 생각하는 개발과정 도중에 정보를 띄움으로써 개발자를 설득하는 사례이다. 애플 XCode에 내장된 정적분석기의 사례도 흥미롭다. 애플은 단순한 경고가 아니라 코드 상에서 버그 발생 경로를 시각적으로 알려줌으로써 인지부하를 덜어주었다. 또한 이 과정을 빌드과정에 통합시킴으로써 알맞은 상황에 정보가 나타나도록 하고 있다.

버그탐지 도구는 개발자를 더 많이 설득할수록 그 영향력이 확대된다. 이제 우리는 단순히 성능을 높이는 것이 아니라, 개발자의 사고 흐름에 자연스럽게 녹아들 방법을 제시해야 한다. 우리의 도구는 개발자가 신뢰하고 기꺼이 사용할 수 있는 유용함을 통해 진정한 가치를 발견해야 한다. 이런 가치를 심분 발휘하여 설득까지 자동화한 도구만이 끝까지 살아남을 것이다. 그렇게 함으로써 우리는 더 많은 개발자들을 돕고 더 안전한 소프트웨어를 통해 세상에 기여할 수 있을 것이다.